

# Evolution, Evolvability, Expression and Engineering

by

Eeshit Dhaval Vaishnav

Bachelor of Technology, Indian Institute of Technology Kanpur

Submitted in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

SEPTEMBER 2022

© 2022 MIT. All rights reserved.

Signature of the Author: \_\_\_\_\_

\_\_\_\_\_  
August 31, 2022

Certified by: \_\_\_\_\_

\_\_\_\_\_  
Prof. Aviv Regev  
Professor of Biology  
Thesis Supervisor

Accepted by: \_\_\_\_\_

\_\_\_\_\_  
Prof. Mary Gehring  
Member, Whitehead Institute  
Director, Biology Graduate Committee

# Summary of Work

## PUBLICATIONS

- **Nature**: [1] (First (and a corresponding) author) ([Nature cover](#)), [2] ([Nature cover](#)), [3], [4]
- **Nature Medicine**: [5] (co-first author)
- **Nature Biotechnology**: [6]
- **Nature Communications**: [7]
- **Cell**: [8]
- **bioRxiv**: [9]

[1] [The evolution, evolvability and engineering of gene regulatory DNA](#)

[2] [A multimodal cell census and atlas of the mammalian primary motor cortex](#)

[3] [A transcriptomic and epigenomic cell atlas of the mouse primary motor cortex](#)

[4] [The human body at cellular resolution: the NIH Human Biomolecular Atlas Program](#)

[5] [Single-cell meta-analysis of SARS-CoV-2 entry genes across tissues and demographics](#)

[6] [Deciphering eukaryotic gene-regulatory logic with 100 million random promoters](#)

[7] [Actomyosin meshwork mechanosensing enables tissue shape to orient cell force](#)

[8] [A Cellular Taxonomy of the Bone Marrow Stroma in Homeostasis and Leukemia](#)

[9] [Reference-based cell type matching of spatial transcriptomics data](#)

**Google Scholar**: <https://scholar.google.com/citations?user=brVs5bAAAAAJ&hl=en>

## PATENT

**insi2vec**, *A framework for inferring from single-cell and spatial multi-omics* ([U.S. Patent Application No. 17/553,691](#)): Methods and systems for determining gene expression profiles and cell identities from multi-omic imaging data

# Acknowledgements

I would like to express my most sincere gratitude to Prof. Aviv Regev (who is the best PhD advisor on the planet), MIT and the Broad Institute for the unbounded opportunities. I would also like to thank them, and my collaborators, thesis committee, colleagues, friends and family for their unwavering support and kindness. The work presented here would be impossible without all of their contributions.

# Evolution, Evolvability, Expression and Engineering

by

Eeshit Dhaval Vaishnav

Submitted on August 31, 2022 in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy at the Massachusetts Institute of Technology

## ABSTRACT

This thesis describes how to build machines (*Engineering*) that answer questions about: (a) *Evolution & Evolvability* and (b) *Expression*.

In the first part of this thesis, I present a framework for understanding and engineering biological sequences, and solving sequence→function problems by building ‘Complete Fitness Landscapes’ in sequence space. This framework for measuring, modelling and designing biological sequences is built around the idea of learning an ‘oracle’ (typically a deep neural network model that takes a sequence as input and predicts its corresponding function) to traverse these ‘Complete Fitness Landscapes’. Here we develop a (promoter sequence)→(gene expression) oracle and use it with our framework to design sequences that demonstrate expression beyond the range of naturally observed sequences. We also show how our framework can be used to detect signatures of selection on a sequence, and to characterize robustness and evolvability.

The second part of this thesis describes two frameworks for inferring from single-cell and spatial gene expression measurements: ATLAS (A Tool for Learning from Atlas-scale Single-cell datasets) and insi2vec (a framework for inferring from spatial multi-omic and imaging measurements).

Thesis Supervisor: Prof. Aviv Regev  
Title: Professor of Biology, MIT  
Core Member, Broad Institute  
Investigator, Howard Hughes Medical Institute  
Head, Roche Genentech Research and Early Development.

# Table of Contents

**Title**.....1

**Summary**.....2

**Acknowledgements** .....3

**Abstract**.....4

**Introduction**.....6

**Part A: *Evolution & Evolvability***.....20

**Part B: *Expression*** .....134

**Discussion**.....159

# Introduction

The goals of the work described in this thesis were to (i) find important questions, (ii) build machines for answering such questions and (iii) construct frameworks for discovering new knowledge. Each of these remain works in progress.

Discovering new knowledge (iii) is almost certainly the most challenging of these goals. This may be because new knowledge seems to emerge spontaneously, almost as a by-product of working on finding important questions and answering them. Reliably conjuring new knowledge likely requires that we build general purpose machines for answering questions.

The significance of finding (and choosing) important questions (i) cannot be overstated. Fortunately, in science, there is a way of doing this reproducibly. If a scientist reads sufficient scientific literature in their areas of interest, the important questions will often become abundantly clear to them. A useful test for a scientist to assess whether a question is ‘important’, is to ask whether the ‘importance’ of these questions becomes immediately clear to everyone else they communicate these questions to. E.g.:

- *Can we predict evolution?*
- *Can we predict gene expression?*
- *Could we discover emergent phenomena from genomes of populations?*
- *Could we understand evolutionary history?*
- *Can we predict future evolvability?*
- *Could we discover principles that, when inductively applied over (evolutionary) time and (sequence) space answer, these questions?*

When working on such audacious grand questions, it can be useful to think of them as ‘homework problems’. This helps convince the scientist that an answer exists, which reflexively helps the process of finding answers and discovering new knowledge. More generally, ‘working backwards from the homework assignments in existing literature and review papers’ can be a useful starting point for thinking about scientific questions.

Once the questions are defined, the work of building machines for answering them (ii) begins. Neural networks are the closest humans have come, to building a general purpose technology for answering questions. I don’t think it is a coincidence (or simply a result of the time period the work presented here was carried out in) that neural networks, which are universal function approximators, ended up becoming pivotal to the work presented in this thesis. Apart from the critical role neural network models played in the design→build→test→learn cycles described here, they were also instrumental in their role as ‘foundation models’ for tasks downstream of this cycle. Much of the work presented here involved the formulation of questions in a way that allowed these machines to learn how to answer the questions posed to them (*machine learning*). Machine learning thus became a critical tool in the process of finding new knowledge (*research*).

This thesis describes our research on building machines (*Engineering*) to answer questions about: (a) *Evolution & Evolvability* and (b) *Expression*.

## EVOLUTION & EVOLVABILITY

In the first part of this thesis, I discuss a framework for thinking about sequence→function problems in terms of ‘Complete Fitness Landscapes’ in sequence space. This framework for measuring, modelling and designing biological sequences is built around the idea of learning an ‘oracle’ (typically a deep neural network model that takes a sequence as input and predicts its corresponding function) to traverse these ‘Complete Fitness Landscapes’. Here we develop a (promoter sequence)→(gene expression) oracle and use it with our framework to design sequences that demonstrated expression beyond the range of naturally observed sequences. We also show how our framework can be used to detect signatures of selection on a sequence, and to characterize robustness and evolvability.

Non-coding regulatory DNA sequences regulate the expression of protein coding sequences of a gene. Changes in regulatory DNA play a major role in the evolution of gene expression<sup>1</sup>. Mutations in *cis*-regulatory elements (CREs) can affect their interactions with transcription factors (TFs), change the timing, location, and level of gene expression, and impact organismal phenotype and fitness<sup>2,3</sup>. While TFs evolve slowly because they each regulate many target genes, CREs evolve much faster and are thought to drive substantial phenotypic variation<sup>7</sup>. Thus, understanding how *cis*-regulatory sequence variation affects gene expression, phenotype and organismal fitness is fundamental to our understanding of regulatory evolution<sup>2</sup>.

A fitness function maps genotypes (which vary through mutations) to their corresponding organismal fitness values (where selection operates)<sup>8</sup>. A complete fitness landscape<sup>9</sup> is defined by a fitness function that maps each sequence in a sequence space to its associated fitness, coupled with an approach for visualizing the sequence space. Partial fitness landscapes have been characterized empirically<sup>4,5,10</sup>, often defining fitness as the maximum growth rate of single-cell



organisms<sup>4,11</sup>. Many recent empirical fitness landscape studies of proteins<sup>12</sup>, adeno-associated viruses<sup>13</sup>, catalytic RNAs<sup>14</sup>, promoters<sup>15</sup>, and TF binding sites<sup>16</sup> have favored molecular activities as fitness proxies because they are less susceptible to experimental biases and measurement noise<sup>17</sup>. In particular, the molecular activity of a promoter sequence as reflected in the expression of the regulated gene has been used to build a ‘promoter fitness landscape’<sup>18</sup>. However, despite advances in high-throughput measurements, empirical fitness landscape studies often sample sequences in the local neighborhood of natural ones and thus remain limited to a tiny subset of the complete sequence space whose size grows exponentially with sequence length ( $4^L$  for DNA or RNA, where  $L$  is the length of sequence)<sup>4-6</sup>.

Understanding the relationship between promoter sequence, expression phenotype, and fitness would allow us to answer fundamental questions<sup>6</sup> in evolution and gene regulation, and provide an invaluable bioengineering tool<sup>6,19</sup>. A model that accurately approximates the relationship between sequence and expression can serve as an “oracle” in evolutionary studies to conduct and interpret *in-silico* experiments<sup>20-23</sup>, predict which regulatory mutations affect expression and fitness (when coupled with expression-to-fitness curves<sup>11</sup>), design or evolve new sequences with desired characteristics, determine how quickly selection achieves an expression optimum, identify signatures of selective pressures on extant regulatory sequences, visualize fitness landscapes and characterize mutational robustness and evolvability<sup>2,4-6,24,25</sup>.

In the first part of this this thesis, we address these long-standing problems by developing a framework for studying regulatory evolution and fitness landscapes based on *Saccharomyces cerevisiae* promoter sequence-to-expression models.

## **EXPRESSION**

The second part of this thesis focusses on approaches for inferring from single-cell and spatial gene expression measurements. Single-cell RNA-sequencing measurements (scRNA-seq)<sup>26</sup> output a ‘feature vector’ for each cell corresponding the cell’s gene expression profile (referred to as its transcriptome). We first describe an approach, that we call ATLAS (A Tool for Learning from Atlas-scale Single-cell datasets), for inferring from large datasets of scRNA-seq measurements (referred to as scRNA-seq atlases). ATLAS allows us to integrate vast scRNA-seq datasets, decipher gene expression programs, predict and prioritize drug targets. ATLAS can also spatially map cell types and predict spatial gene expression patterns. We demonstrate the applications of ATLAS in the context of COVID-19<sup>27</sup>.

***ATLAS:*** The Coronavirus Disease 2019 (COVID-19) caused by the SARS-CoV-2 virus has led to significant global morbidity and mortality<sup>28</sup>. With no standard cures or vaccines available yet to contain and prevent the disease, there is an emergent need to understand COVID-19 onset, progression and the underlying disease mechanisms at the molecular level to enable discovery of pathways and targets for treatments. COVID-19 starts with the infection or entry of SARS-CoV-2 into human cells, primarily in the upper respiratory system. Subsequent replication and spread of the virus to other human cells, mounts inflammation and immune responses leading to the widespread clinical pathologies including pneumonia in moderate cases, and acute respiratory distress and death in severe cases. A key to understanding disease mechanisms in turn lies in identifying molecular changes in virus infected cells observed in COVID-19 patients. The genetic code in each of these cells is organized into (over 20000) units called genes, which are converted into functional gene products like proteins through a process referred to as gene

expression. Gene expression is regulated by gene regulatory networks that involve complex, non-linear and combinatorial interactions between genes<sup>29</sup>. Human phenotypes, such as "healthy" or "diseased" states, are induced by the expression of these genes in relevant virus-infected and bystander responsive cells. Single-cell RNA-sequencing (scRNA-seq) is an experimental technology that allows us to measure gene expression in such individual cells isolated from tissues. The output of every scRNA-seq experiment is a high-dimensional, sparse gene expression vector for each isolated cell whose dimensions correspond to the expression levels of genes within the cell. Machine learning methods are then applied to the output of scRNA-seq experiments for dimensionality reduction<sup>30</sup>, unsupervised clustering<sup>31</sup> and gene expression program inference<sup>32</sup>. The construction of single-cell atlases involves multiple scRNA-seq experiments run on tissues isolated from a diverse set of individuals, who are dispersed along a continuum of phenotypes. These experiments are conducted under varying experimental conditions and run on a wide range of technological platforms in different laboratories. This introduces complex, non-linear variability in measurements made across these experimental domains and hinders our ability to make effective comparisons between measurements made across domains (e.g. comparing differences in gene expression programs between healthy and diseased individuals). These domain-specific effects are an impediment to the process of making fundamental biological discoveries and to translational applications such as the identification of targets for therapeutic interventions to treat diseases like COVID-19. Extensive previous work on the identification of drug targets has focused on convolutional and graph neural networks for modeling protein structures<sup>33</sup> and molecular interactions<sup>34-35</sup> between drugs and their putative targets. But, a critical upstream step in drug development is the identification of these biological

targets. Single-cell atlases have the potential to transform the process of identification of genes involved in the modulation of disease phenotype.

There is currently no principled, generalizable approach for identifying drug targets by inferring gene expression programs from single-cell atlases to treat human diseases. To address this, one would need to adequately address the two-fold challenge in single-cell atlas analysis described above : (i) the combinatorial and non-linear effects of gene expression on phenotypes and (ii) the variability in measurements across experimental domains.

The task of remedying the variability in scRNA-seq measurements across experimental domains is a type of domain adaptation<sup>36</sup> problem where the objective is to learn domain-invariant feature representations of scRNA-seq data. Domain-invariant feature representations are central to a large body of work on domain adaptation<sup>37-41</sup>. In the context of scRNA-seq, domain adaptation is referred to as data-integration and is defined as the process of generating an internally- consistent version of the data<sup>42</sup> across these measurement domains. Existing methods carry out this scRNA-seq data-integration task by either operating directly on the full gene expression vectors<sup>43-44</sup> or operating on representations derived from them like nearest-neighbor graphs<sup>45-47</sup> and learned low-dimensional embeddings<sup>48-51</sup>. These methods employ a broad range of statistical and machine learning techniques including panoramic stitching<sup>52</sup>, canonical correlation analysis<sup>53</sup>, non-negative matrix factorization<sup>54</sup> and perturbation modeling<sup>49</sup>. Consequently, comprehensive metrics<sup>55</sup> have been developed for evaluating the efficacy of these domain integration methods in ameliorating domain-specific effects while conserving biological information. Benchmarking studies<sup>42,56</sup> for the domain integration task using these metrics demonstrate the need and room for vast improvements.

Additionally, the domain-invariant representations learned by many of these methods<sup>46,54,57-58</sup> cannot be mapped back to the input gene expression vectors which significantly restricts the applicability of such representations to biologically meaningful tasks.

On the other hand, an autoencoder (AE)<sup>59-60</sup> has the ability to learn feature representations that can be mapped back to the input space using a pair of functions: an encoder for transforming the input into this representation and a decoder for reconstructing the input from this representation. AEs can learn domain-invariant representations by minimizing the domain discrepancy between features learned by the encoder using a domain regularizer or by encouraging domain confusion using an adversarial objective<sup>41,61</sup>. Maximum Mean Discrepancy (MMD)<sup>62</sup>, a widely used regularizer for aligning the first two moments of a distribution, has recently been applied to the scRNA-seq data- integration problem<sup>49-51</sup>. However, scRNA-seq data<sup>63</sup> and their latent representations learned by an AE follow non-Normal distributions. First and second order moment matching methods do not suffice for minimizing the domain discrepancy in many real-world unsupervised domain adaptation problems and so methods for matching higher-order statistics have been proposed<sup>64-66</sup> for the task of transferring labels from a labelled source domain to an unlabelled target domain. However, an approach for incorporating higher order moment-matching regularizers to the multi-target domain adaptation<sup>67</sup> task in general, and the scRNA-seq data integration problem, in particular is lacking.

We hypothesize that domain-invariant feature representations of scRNA-seq data can help address the pressing need for identification of human disease drug targets when used in tandem with

modern feature importance methods to account for the combinatorial and non-linear effects of gene-expression on phenotype. This forms the basis of ATLAS.

*insi2vec*: scRNA-seq captures cell-intrinsic information but the process of making these measurements, which involves the dissociation of tissues, leads to a loss of spatial and cell-extrinsic contextual information. Spatial transcriptomic measurements overcome these limitations, but come with their own sets of trade-offs<sup>68</sup>. We present *insi2vec*<sup>69</sup>, a framework for inferring from single-cell and spatial multi-omics, to address these challenges. *insi2vec*, consists of: (i) a spatio-transcriptomic definition of cell identity using cell intrinsic and cell extrinsic features, and (ii) methods for predicting spatial gene expression patterns from (ii-a) single-cell RNA-sequencing measurements and (ii-b) histology. We demonstrate the applications of *insi2vec* to cancer and brain research.

## References

1. Wittkopp, P. J. & Kalay, G. Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence. *Nat. Rev. Genet.* **13**, 59–69 (2011).
2. Hill, M. S., Vande Zande, P. & Wittkopp, P. J. Molecular and evolutionary processes generating variation in gene expression. *Nature Reviews Genetics* 1–13 (2020).
3. Fuqua, T. *et al.* Dense and pleiotropic regulatory information in a developmental enhancer. *Nature* 1–5 (2020).
4. de Visser, J. A. G. M. & Krug, J. Empirical fitness landscapes and the predictability of evolution. *Nat. Rev. Genet.* **15**, 480–490 (2014).
5. Kondrashov, D. A. & Kondrashov, F. A. Topological features of rugged fitness landscapes in sequence space. *Trends in Genetics* **31**, 24–33 (2015).
6. de Visser, J. A. G. M., Elena, S. F., Fragata, I. & Matuszewski, S. The utility of fitness landscapes and big data for predicting evolution. *Heredity (Edinb)* **121**, 401–405 (2018).
7. Weirauch, M. T. & Hughes, T. R. Conserved expression without conserved regulatory sequence: the more things change, the more they stay the same. *Trends Genet.* **26**, 66–74 (2010).
8. Orr, H. A. The genetic theory of adaptation: a brief history. *Nature Reviews Genetics* **6**, 119–127 (2005).
9. Weinreich, D. M., Lan, Y., Wylie, C. S. & Heckendorn, R. B. Should evolutionary geneticists worry about higher-order epistasis? *Current Opinion in Genetics & Development* **23**, 700–707 (2013).
10. Venkataram, S. *et al.* Development of a Comprehensive Genotype-to-Fitness Map of Adaptation-Driving Mutations in Yeast. *Cell* **166**, 1585-1596.e22 (2016).
11. Keren, L. *et al.* Massively Parallel Interrogation of the Effects of Gene Expression Levels on Fitness. *Cell* **166**, 1282-1294.e18 (2016).
12. Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature* **533**, 397–401 (2016).
13. Ogden, P. J., Kelsic, E. D., Sinai, S. & Church, G. M. Comprehensive AAV capsid fitness landscape reveals a viral gene and enables machine-guided design. *Science* **366**, 1139–1143 (2019).
14. Pitt, J. N. & Ferré-D’Amaré, A. R. Rapid construction of empirical RNA fitness landscapes. *Science* **330**, 376–379 (2010).
15. Shultzaberger, R. K., Malashock, D. S., Kirsch, J. F. & Eisen, M. B. The Fitness Landscapes of cis-Acting Binding Sites in Different Promoter and Environmental Contexts. *PLOS Genetics* **6**, e1001042 (2010).
16. Mustonen, V., Kinney, J., Callan, C. G. & Lässig, M. Energy-dependent fitness: a quantitative model for the evolution of yeast transcription factor binding sites. *Proc. Natl. Acad. Sci. U.S.A.* **105**, 12376–12381 (2008).
17. Hartl, D. L. What Can We Learn From Fitness Landscapes? *Curr Opin Microbiol* **0**, 51–57 (2014).

18. Otwinowski, J. & Nemenman, I. Genotype to phenotype mapping and the fitness landscape of the E. coli lac promoter. *PLoS ONE* **8**, e61570 (2013).
19. Sinai, S. & Kelsic, E. D. A primer on model-guided exploration of fitness landscapes for biological sequence design. *arXiv:2010.10614 [cs, q-bio]* (2020).
20. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods* **12**, 931–934 (2015).
21. Avsec, Ž. *et al.* Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat. Genet.* **53**, 354–366 (2021).
22. Shrikumar, A., Greenside, P. & Kundaje, A. Learning Important Features Through Propagating Activation Differences. *arXiv [cs.CV]* (2017).
23. Avsec, Ž. *et al.* Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* **18**, 1196–1203 (2021).
24. Fragata, I., Blanckaert, A., Louro, M. A. D., Liberles, D. A. & Bank, C. Evolution in the light of fitness landscape theory. *Trends in Ecology & Evolution* **34**, 69–82 (2019).
25. Payne, J. L. & Wagner, A. The causes of evolvability and their evolution. *Nature Reviews Genetics* **20**, 24–38 (2019).
26. Tanay, A., Regev, A. Scaling single-cell genomics from phenomenology to mechanism. *Nature* **541**, 331–338 (2017).
27. Muus, C., Luecken, M.D., Eraslan, G. *et al.* Single-cell meta-analysis of SARS-CoV-2 entry genes across tissues and demographics. *Nat Med* **27**, 546–559 (2021).
28. Mortality analyses. <https://coronavirus.jhu.edu/data/mortality>. Accessed: 2020-6-4.
29. Ido Amit, Manuel Garber, Nicolas Chevrier, Ana Paula Leite, Yoni Donner, Thomas Eisenhaure, Mitchell Guttman, Jennifer K Grenier, Weibo Li, Or Zuk, Lisa A Schubert, Brian Birditt, Tal Shay, Alon Goren, Xiaolan Zhang, Zachary Smith, Raquel Deering, Rebecca C McDonald, Moran Cabili, Bradley E Bernstein, John L Rinn, Alex Meissner, David E Root, Nir Hacohen, and Aviv Regev. Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses. *Science*, 326(5950):257–263, October 2009.
30. Pierson, E., Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol* **16**, 241 (2015).
31. Wang, B., Zhu, J., Pierson, E. *et al.* Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat Methods* **14**, 414–416 (2017).
32. Dylan Kotliar, Adrian Veres, M Aurel Nagy, Shervin Tabrizi, Eran Hodis, Douglas A Melton, and Pardis C Sabeti. Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq. *Elife*, 8, July 2019.
33. Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Židek, Alexander W R Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, January 2020.



34. Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. March 2017.
35. Joseph Gomes, Bharath Ramsundar, Evan N Feinberg, and Vijay S Pande. Atomic convolutional networks for predicting Protein-Ligand binding affinity. March 2017.
36. Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1):151–175, May 2010.
37. Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. January 2013.
38. Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, February 2013.
39. Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. January 2019.
40. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for Large-Scale sentiment classification: A deep learning approach. January 2011.
41. Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.
42. M D Luecken, M Büttner, K Chaichoompu, A Danese, M Interlandi, M F Mueller, D C Strobl, L Zappia, M Dugas, M Colomé-Tatché, and F J Theis. Benchmarking atlas-level data integration in single-cell genomics. May 2020.
43. W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, January 2007.
44. Uri Shaham, Kelly P Stanton, Jun Zhao, Huamin Li, Khadir Raddassi, Ruth Montgomery, and Yuval Kluger. Removal of batch effects using distribution-matching residual networks. *Bioinformatics*, 33(16):2539–2546, August 2017.
45. Krzysztof Polan’ski, Matthew D Young, Zhichao Miao, Kerstin B Meyer, Sarah A Teichmann, and Jong-Eun Park. BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics*, 36(3):964–965, 08 2019.
46. Nikolas Barkas, Viktor Petukhov, Daria Nikolaeva, Yaroslav Lozinsky, Samuel Demharter, Konstantin Khodosevich, and Peter V Kharchenko. Joint analysis of heterogeneous single-cell RNA-seq dataset collections. *Nat. Methods*, 16(8):695–698, August 2019.
47. Laleh Haghverdi, Aaron T L Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.*, 36(5):421–427, June 2018.
48. Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12):1053–1058, December 2018.

49. Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian Theis, and F. Wolf. Conditional out-of-sample generation for unpaired data using trvae. *arXiv*, 10 2019.
50. Valentine Svensson, Adam Gayoso, Nir Yosef, and Lior Pachter. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics*, March 2020.
51. Matthew Amodio, David van Dijk, Krishnan Srinivasan, William S Chen, Hussein Mohsen, Kevin R Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy, Anita Desai, V Ravi, Priti Kumar, Ruth Montgomery, Guy Wolf, and Smita Krishnaswamy. Exploring single-cell data with deep multitasking neural networks. *Nat. Methods*, 16(11):1139–1145, November 2019.
52. Bonnie Berger Brian Hie, Bryan Bryson. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature Biotechnology*, 37(June):685–691, 2019.
53. Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck, 3rd, Yuhao Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of Single-Cell data. *Cell*, 177(7):1888–1902.e21, June 2019.
54. Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-Cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887.e17, June 2019.
55. Maren Büttner, Zhichao Miao, F Alexander Wolf, Sarah A Teichmann, and Fabian J Theis. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods*, 16(1):43–49, January 2019.
56. Ruben Chazarra-Gil, Stijn van Dongen, Vladimir Yu Kiselev, and Martin Hemberg. Flexible comparison of batch correction methods for single-cell RNA-seq using BatchBench. May 2020.
57. Krzysztof Polanski, Matthew D Young, Zhichao Miao, Kerstin B Meyer, Sarah A Teichmann, and Jong-Eun Park. BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics*, 36(3):964–965, February 2020.
58. Korsunsky, I., Millard, N., Fan, J. et al. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat Methods* 16, 1289–1296 (2019).
59. Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In J D Cowan, G Tesauro, and J Alspecter, editors, *Advances in Neural Information Processing Systems 6*, pages 3–10. Morgan-Kaufmann, 1994.
60. Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, August 2013.
61. Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. February 2017.
62. Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(Mar):723–773, 2012.
63. Valentine Svensson. Droplet scRNA-seq is not zero-inflated. *Nat. Biotechnol.*, 38(2):147–150, February 2020.

64. Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for Domain-Invariant representation learning. February 2017.
65. Werner Zellinger, Bernhard A Moser, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Robust unsupervised domain adaptation for neural networks via moment alignment. *Inf. Sci.*, 483:174–191, May 2019.
66. Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. HoMM: Higher-order moment matching for unsupervised domain adaptation. December 2019.
67. Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised Multi-Target domain adaptation: An information theoretic approach. October 2018.
68. Rao, A., Barkley, D., França, G.S. *et al.* Exploring tissue architecture using spatial transcriptomics. *Nature* **596**, 211–220 (2021).
69. Methods and systems for determining gene expression profiles and cell identities from multi-omic imaging data. U.S. Patent No. US20220180975A1, 2022.



## **Part A:**

### *Evolution & Evolvability*

# The evolution, evolvability and engineering of gene regulatory DNA

Eeshit Dhaval Vaishnav<sup>1,2,12</sup>✉, Carl G. de Boer<sup>3,4,12</sup>✉, Jennifer Molinet<sup>5,6</sup>, Moran Yassour<sup>4,7,8</sup>, Lin Fan<sup>2</sup>, Xian Adiconis<sup>4,9</sup>, Dawn A. Thompson<sup>2</sup>, Joshua Z. Levin<sup>4,9</sup>, Francisco A. Cubillos<sup>5,6</sup> & Aviv Regev<sup>4,10,11</sup>✉



**Paper:** <https://doi.org/10.1038/s41586-022-04506-6>

**Code:** <https://github.com/1edv/evolution>

**Data:** <https://bit.ly/EvolutionZenodo>

**App:** <https://1edv.github.io/evolution/>

**Nature Cover:** <https://www.nature.com/nature/volumes/603/issues/7901>

**Contribution:** First (and a corresponding) author

# The evolution, evolvability, and engineering of gene regulatory DNA

Eeshit Dhaval Vaishnav<sup>1,2,11§</sup>, Carl G. de Boer<sup>3,8,11§</sup>, Jennifer Molinet<sup>4,5</sup>, Moran Yassour<sup>6,7,8</sup>, Lin Fan<sup>2</sup>, Xian Adiconis<sup>8,9</sup>, Dawn A. Thompson<sup>2</sup>, Joshua Z. Levin<sup>8,9</sup>, Francisco A. Cubillos<sup>4,5</sup>, & Aviv Regev<sup>8,10,12§</sup>

<sup>1</sup>Massachusetts Institute of Technology, Cambridge, MA 02139, USA

<sup>2</sup>Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

<sup>3</sup>School of Biomedical Engineering, University of British Columbia, Vancouver, BC, V6T 1Z3, Canada

<sup>4</sup>Universidad de Santiago de Chile, Facultad de Química y Biología, Departamento de Biología, Santiago, 9170022, Chile.

<sup>5</sup>ANID – Millennium Science Initiative Program - Millennium Institute for Integrative Biology (iBio). Santiago, 7500574, Chile.

<sup>6</sup>Faculty of Medicine, The Hebrew University of Jerusalem, Jerusalem 91121, Israel

<sup>7</sup>The Rachel and Selim Benin School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, Israel

<sup>8</sup>Klarman Cell Observatory, Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

<sup>9</sup>Stanley Center for Psychiatric Research, Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

<sup>10</sup>Howard Hughes Medical Institute and Koch Institute of Integrative Cancer Research, Department of Biology, Massachusetts Institute of Technology, Cambridge, MA 02140 USA

<sup>11</sup>These authors contributed equally

<sup>12</sup>Current address: Genentech, 1 DNA Way, South San Francisco, CA, USA

<sup>§</sup>Correspondence should be addressed to: [edv@mit.edu](mailto:edv@mit.edu), [carl.deboer@ubc.ca](mailto:carl.deboer@ubc.ca), [aviv.regev.sc@gmail.com](mailto:aviv.regev.sc@gmail.com)

## SUMMARY

Mutations in non-coding regulatory DNA sequences can alter gene expression, organismal phenotype, and fitness<sup>1-3</sup>. Constructing complete fitness landscapes, mapping DNA sequences to fitness, is a long-standing goal in biology, but has remained elusive because it is challenging to generalize reliably to vast sequence spaces<sup>4-6,8-17</sup>. Here, we construct sequence-to-expression models that capture fitness landscapes and use them to decipher principles of regulatory evolution<sup>7</sup>. Using millions of randomly-sampled promoter DNA sequences<sup>18</sup> and their measured expression levels in the yeast *Saccharomyces cerevisiae*, we learn deep neural network models that generalize with excellent prediction performance, and enable sequence design for expression engineering<sup>19</sup>. Using our models, we study expression divergence under genetic drift and strong-selection weak-mutation regimes<sup>20-23</sup> to find that regulatory evolution is rapid and subject to diminishing returns epistasis, that conflicting expression objectives in different environments constrain expression adaptation, and that stabilizing selection on gene expression leads to the moderation of regulatory complexity. We present an approach for using our models to detect signatures of selection on expression from natural variation in regulatory sequences and use it to discover an instance of convergent regulatory evolution. We assess mutational robustness, finding that regulatory mutation effect sizes follow a power law, characterize regulatory evolvability, visualize promoter fitness landscapes, discover evolvability archetypes and highlight the mutational robustness of natural regulatory sequence populations<sup>24-25</sup>. Our work provides a general framework for addressing fundamental questions in regulatory evolution.



## RESULTS

### Models predict expression from sequence

We begin by building models that predict gene expression given an 80 bp promoter DNA sequence. To train these models, we measure the expression driven by promoter sequences using an approach we previously described<sup>26</sup>, where 80 bp of DNA are embedded within a promoter construct and the associated expression is assayed in the *S. cerevisiae* (**Methods**). We clone promoter sequences into an episomal low copy number YFP expression vector, transform them into yeast, culture the yeast in the desired media, sort the yeast into 18 expression bins, and sequence the promoters present from the yeast in each bin to estimate expression (**Methods** and **Supplementary Information**). To avoid biases<sup>5</sup> towards extant sequences, we measured the expression of 80 bp random DNA sequences, where each base is randomly sampled from the four bases. For training data, we measured each of >30 million sequences in complex media (YPD, **Methods**) and >20 million sequences in defined media (SD-Ura, synthetic defined lacking uracil). Using the resulting pairs of sequences and measured YFP expression levels, we trained convolutional neural network models (“convolutional models”) that predict expression from sequence in each medium (**Methods**).

To show that the learned convolutional models generalize to new sequences, we predicted the expression for several sets of test sequences not seen during model training, and compared them to their experimentally measured levels (**Methods**). For these test sequences, we quantified expression in independent experiments using the same experimental approach and in the same media. Our convolutional models had excellent prediction performance on native yeast promoter

test sequences (Pearson's  $r = 0.960$ ,  $P < 5 \cdot 10^{-324}$ ,  $n=61,150$ ; **Fig. 1b**), and on multiple other test sets in both complex and defined media (**Extended Data Fig. 1**).

These results represent a ~45% decrease in error compared to the performance of biochemical models we previously<sup>26</sup> trained on the same data (complex media; native yeast promoter test sequences; **Supplementary Notes and Methods**). Other published genomic model architectures adapted to and trained using our data also had excellent performance (**Supplementary Fig. 4a**), highlighting the predictive power of deep neural network models trained using our large-scale data. Finally, the expression measurements were highly correlated for the same sequences between the two media (Pearson's  $r = 0.978$ , **Extended Data Fig. 2a**) and models trained on defined medium predicted expression in complex medium well (Pearson's  $r = 0.966$ , **Extended Data Fig. 2b**). However, for some sequences we expect differences between growth conditions (below).

### **Models enable expression engineering**

We leveraged the high predictive performance of our convolutional models for a synthetic biology application of gene expression engineering, by using model predictions as a 'fitness function' for genetic algorithms (GA) to design sequences with extreme expression values. We initialized the GA with a population of 100,000 randomly-generated samples from the sequence space, and simulated 10 generations to maximize (or minimize) the expression output from the convolutional model (**Methods**). We then synthesized the 500 sequences with the top predicted maximum (or minimum) expression levels and tested them experimentally. The GA-designed sequences drove, on average, more extreme expression than >99% of native sequences (99.6% for high expressing; 99.3% for low), with ~20% of designed sequences yielding more extreme expression than any

native sequence tested (23.5% for high; 18.4% for low) (**Fig. 1c**). Thus, our sequence-to-expression model can be used for gene expression engineering.

### **Expression diverges under genetic drift**

We next assessed the evolutionary malleability of expression under different evolutionary scenarios: random genetic drift, stabilizing selection, and directional selection for extreme expression levels (**Fig. 2**). In each case, we first simulated the scenario, using our convolutional model to predict the expression for each sequence, and then tested the model's evolved sequences experimentally, where possible (**Methods**).

We first simulated random genetic drift of regulatory sequences, with no selection on expression levels. We randomly introduced a single mutation in each random starting sequence, repeated this process for multiple consecutive generations, and used our convolutional model to predict the difference in expression between the mutated sequences in each trajectory relative to the corresponding starting sequence (**Fig. 2a-c**). Expression levels diverged as the number of mutations increased, with 32 mutations in the 80 bp region resulting in nearly as different expression from the original sequence as two unrelated sequences (**Fig. 2b**). We validated our results experimentally by synthesizing sequences with zero to three random mutations and measuring their expression in our assay (**Methods**). The experimental measurements closely matched our predictions in both complex (**Fig. 2c**) and defined (**Extended Data Fig. 1e**) media, both in expression change (Pearson's  $r$ : 0.869 and 0.847, respectively; **Extended Data Fig. 1h,i**) and level (Pearson's  $r$ : 0.973 and 0.963 respectively; **Extended Data Fig. 1l,m**).

## Stabilizing selection tempers complexity

Although gene regulatory networks often appear to be highly interconnected<sup>26,27</sup>, the sources of this regulatory complexity and how it changes with the turnover of regulatory mechanisms<sup>28</sup> remain unclear. We used our model to study the evolution of regulatory complexity in the context of stabilizing selection, which favors the maintenance of existing expression levels. We first quantified regulatory complexity, defined as 1 minus the Gini coefficient (a measure of inequality of continuous values within a population) of TF regulatory interaction strengths. For this, we used an interpretable biochemical model we previously developed<sup>26</sup> (**Methods**) because it has parameters that explicitly correspond to TFs, and we can directly query their contributions to model predictions. Next, starting with native sequences whose regulatory complexity is either extremely high (many TFs with similar contributions to expression) or low (few TFs contribute disproportionately to expression) and spanning a range of expression levels, we introduced single mutations into each starting native sequence for each of 32 consecutive generations, identified the sequences that conserved the original expression level using the convolutional model, and selected one of them at random for the next generation. We then assessed the regulatory complexity of the evolved sequences.

As random mutations accumulated, the regulatory complexity of sequences starting at both complexity extremes shifted towards moderate complexities (**Fig. 2d**, rightmost blue and orange), closer to the averages for both random and native sequences (**Fig. 2d**, greys). This suggests that stabilizing selection on expression leads to a moderation of regulatory complexity, resulting from gradual drift in the roles of the different regulators, such as an increase in complexity due to a decrease in the relative contribution of one predominant TF (*e.g.* Abf1p for *AIF1*), or a decrease

in complexity through smaller changes in a much larger number of sites (*e.g.* *YDR476C*; **Supplementary Fig. 8**). The overall distribution of regulatory complexity of native yeast promoters is similar to that of random sequences (**Fig. 2d**, grey boxes), suggesting that there is little selection on the regulatory complexity of native sequences in a single environment.

### **Strong selection rapidly finds extrema**

To study the impact of directional selection on expression, we simulated the strong-selection weak-mutation (SSWM) regime<sup>29</sup> (**Fig. 2e**, **Methods**), where each mutation is either beneficial or deleterious (strong selection, with mutations surviving drift and fixing in an asexual population), and mutation rates are low enough to only consider single base substitutions during adaptive walks (weak mutation). Starting with a set of native promoter sequences, at each iteration (generation), for a given starting sequence of length  $L$ , we considered all of its  $3L$  single-base mutational neighbors, used our convolutional model to predict their expression, and took the sequence with the largest increase (or separately, decrease) in expression at each iteration (generation) as the starting sequence for the next generation (**Fig. 2e**, **Methods**).

Sequences that started with diverse initial expression levels rapidly evolved to high (or separately, low) expression, with the vast majority evolving close to saturating extreme levels within 3-4 mutations in both the complex (**Fig. 2f**) and defined (**Extended Data Fig. 1f**) media. Sequences took diverse paths to evolve either high or low expression (**Supplementary Fig. 7**). We validated these trajectories experimentally for select series of sequences (**Fig. 2g**, **Extended Data Fig. 1g**), measuring the expression driven by synthesized sequences from several generations along

simulated mutational trajectories for complex media (10,322 sequences from 877 trajectories) and defined media (6,304 sequences from 637 trajectories). We observed extreme expression within 3-4 mutational steps, with high agreement between measured and predicted expression change (**Extended Data Fig. 1j,k**; Pearson's  $r$ : 0.977 and 0.948, respectively) and expression levels (**Extended Data Fig. 1n,o**; Pearson's  $r$ : 0.980 and 0.963) along the trajectories in both complex and defined media. Thus, *cis*-regulatory sequence evolution is rapid and subject to diminishing returns epistasis<sup>30</sup>.

### **Opposing objectives constrain adaptation**

In contrast to the rapid evolution towards expression extremes, we found that evolution to satisfy two opposing expression requirements (one in each growth media) was more constrained. A concrete example is the expression of the *URA3* gene: organismal fitness *increases* with increased *URA3* expression in defined media lacking uracil, because Ura3p is required for uracil biosynthesis, but fitness *decreases* with increased *URA3* expression in complex media containing 5-FOA due to Ura3p-mediated conversion of 5-FOA to toxic 5-fluorouracil (**Extended Data Fig. 2c**). To study this regime<sup>31</sup>, we started with a set of native promoter sequences (and separately, a set of random sequences) and used the convolutional model to simulate SSWM trajectories (**Methods**) that maximize the *difference* in expression between the two media (defined and complex). While the difference in expression increased with each generation (**Extended Data Fig. 2d,e**), the vast majority of sequences achieved neither the maximal nor the minimal expression in either condition after 10 generations (**Fig. 2h**, **Extended Data Fig. 2f**), for both native and random starting sequences. The evolved sequences became enriched for motifs for TFs involved in nutrient sensing and metabolism, compared to the starting sequences (**Extended Data Fig. 2g**), suggesting

that the model is taking advantage of subtle differential activity of certain regulators between the two conditions to evolve condition specificity. Thus, while evolving a sequence to achieve a single expression optimum requires very few mutations, encoding multiple opposing objectives in the same sequence is more difficult, limiting expression adaptation.

### **Transformers enable inference at scale**

We next turned to the evolution and evolvability of regulatory sequences in extant strains and species. This required us to predict expression for billions of sequences and, although our convolutional model had excellent predictive power, our implementation was limited in its scalability and incompatible with the Tensor Processing Units (TPUs), available to us for larger-scale computational tasks (**Methods**). To enable large-scale expression prediction, we developed “transformer” models that used transformer encoders<sup>32</sup> with other building blocks attempting to implicitly capture known aspects of regulation<sup>33</sup> (**Methods, Supplementary Fig. 12**). The transformer models had ~20x fewer parameters than the convolutional models (**Methods, Supplementary Information**), predicted expression as well as the convolutional models (**Extended Data Fig. 3**), and better captured the propensity for expression to plateau under SSWM (**Supplementary Fig. 19**). The convolutional and transformer models had highly correlated predictions in both media (**Supplementary Fig. 4e-h**, Pearson’s  $r=0.967-0.985$ ), and yielded equivalent conclusions from the analyses of genetic drift, directional selection and conflicting objectives (**Extended Data Fig. 3, Supplementary Fig. 17-18**).

## The Expression Conservation Coefficient

We applied our sequence-to-expression transformer model to detect evidence of selective pressures on natural regulatory sequences, inspired by the way in which the ratio of non-synonymous (“non-neutral”) to synonymous (“neutral”) substitutions ( $d_N/d_S$ ) in protein coding sequences is used estimate the strength and mode of natural selection<sup>34</sup>. By analogy<sup>2,35</sup>, for regulatory sequences<sup>2</sup>, we used the transformer model to quantitatively assess the impact of naturally occurring regulatory genetic variation on expression, compared to that expected with random mutations, and summarized this with an Expression Conservation Coefficient (ECC) (**Methods**). To compute the ECC, we compared, for each gene’s promoter, the standard deviation of the expression distribution predicted by the transformer model for a set of naturally varying orthologous promoters ( $\sigma_B$ ) to the standard deviation of the expression distribution predicted for a matched set of random variation introduced to that promoter ( $\sigma_C$ ; related to the mutational variance<sup>36</sup>; **Fig. 3a**). We define the ECC for a gene as  $\log(\sigma_C/\sigma_B)$ , such that a positive ECC indicates stabilizing selection on expression (lower variance in native sequences than expected by chance), a negative ECC indicates diversifying (disruptive) selection or local adaptation (greater variance in native sequences), and values near 0 suggest neutral drift.

We calculated the ECC for 5,569 *S. cerevisiae* genes using the natural variation observed across over 4.73 million orthologous promoter sequences from the 1,011 *S. cerevisiae* isolates<sup>37</sup> in the -160 to -80 regions (with respect to the Transcription Start Site (TSS)), a critical location for TF binding<sup>38</sup> and determinant of promoter activity<sup>26</sup> (**Fig. 3a,b, Supplementary Table 1**), using our transformer model to predict the expression for each sequence. To assess the robustness of the ECC values, we recomputed the ECC using multiple published sequence-to-expression model



architectures that we adapted and trained using our data and found that models with similarly high predictive power resulted in similar ECC values (**Supplementary Fig. 4b-d, 5g**).

Over 70% of promoters had positive ECCs, suggesting stabilizing selection (and conserved expression) (binomial test  $P < 10^{-215}$ ) (**Fig. 3b**), consistent with previous reports based on direct measurements of gene expression<sup>39</sup>. Genes with high ECCs were enriched in highly-conserved core cellular processes (*e.g.*, RNA and protein metabolism) (**Fig. 3b, Supplementary Table 2**), and those with low ECCs were most enriched in processes related to carboxylic acid and alcohol metabolism (**Fig. 3b, Supplementary Table 2**), potentially reflecting adaptation of fermentation genes to the diverse environments of these isolates<sup>37</sup>.

### **The ECC discovers convergent evolution**

A striking example of predicted positive selection is the promoter of *CDC36* (*NOT2*; ECC= -2.138, **Fig. 3b**), which has common natural alleles with either low or high (predicted) expression across the isolates (**Fig. 3c**). Analysis of *CDC36* promoter sequences (**Methods**) suggests that low-expression evolved at least twice independently, resulting in two distinct variants with reduced expression (**Fig. 3c**, allele 1 and 2). Interrogation with the biochemical model<sup>26</sup> to identify factors impacting these expression differences (**Extended Data Fig. 4a**) suggested that both low-expression alleles are explained by disruption of the same binding site for Upc2p, an ergosterol sensing TF (**Fig. 3c**). To validate this, we restored the putative Upc2p binding site in a strain (WE), where it is otherwise disrupted, and measured expression levels by qPCR and growth upon changing carbon source (**Methods**). Restoration of the Upc2p binding site increased actual

expression, confirming the model's prediction (Pearson's  $r=0.96$ ,  $p=0.039$ ,  $n=4$ ; **Fig. 3d**). We hypothesized that these variants could alter the rate of transcriptional reprogramming when changing environments via Cdc36p-regulated mRNA turnover<sup>40</sup>. Indeed, restoration of the Upc2 binding site reduced the strains' lag time to growth when switching carbon sources (**Fig. 3d**, right; **Methods**), and they grew to a higher culture density (**Supplementary Fig. 10**). Thus, convergent evolution of the *CDC36* promoter, discovered using the ECC, independently produced two alleles that result in similar perturbations to TF binding, expression, and growth.

### **ECC vs. cross-species RNAseq and fitness**

ECC values were consistent with expression conservation as measured for yeast orthologs across clades at short (*Saccharomyces*), medium (Ascomycota), or long (mammals) evolutionary scales (**Extended Data Fig. 4b**). In *Saccharomyces*, 1:1 orthologs with conserved expression levels across species (as measured by RNA-seq<sup>41</sup>) had significantly higher ECC (computed from the 1,011 yeast isolates) than genes whose expression was not conserved (two-sided Wilcoxon rank-sum  $P = 3.1 \times 10^{-4}$ , **Extended Data Fig. 4b, bottom left, Methods**). Next, we performed RNA-seq across 11 Ascomycota yeast species (**Methods**), finding that 1:1 orthologs with conserved expression across Ascomycota had significantly higher ECC values (**Extended Data Fig. 4b, bottom center**,  $P = 1.16 \times 10^{-6}$ ). Finally, the 1:1 orthologs of genes with high ECC values in the 1,011 *S. cerevisiae* isolates also had more conserved expression within mammals<sup>42</sup> (**Extended Data Fig. 4b, bottom right**,  $P = 1.07 \times 10^{-4}$ , **Methods**). Thus, while 1:1 yeast-mammal orthologs are likely critical to an organism's fitness, only a subset of these may be under stabilizing selection on expression, and this subset tends to be under such selection in both yeasts and mammals. Thus,

the ECC quantifies stabilizing selection on expression in yeast and may predict stabilizing selection on orthologs' expression in other species.

Genes with higher ECCs also had a stronger effect on fitness in *S. cerevisiae* upon changing their expression level. We interrogated the total variation of previously measured expression-to-fitness curves<sup>11</sup> to calculate a 'fitness responsivity' score that captures the dependence of fitness on expression (**Extended Data Fig. 5, Methods**). Fitness responsivity was significantly positively correlated with the ECC (**Supplementary Fig. 2e**,  $P = 0.003$ , Spearman  $\rho = 0.326$ ). Fitness responsivity was not associated with regulatory sequence divergence *per se* across the promoter sequence (as estimated by mean Hamming distance among orthologous promoters, **Methods, Supplementary Fig. 2d**,  $P = 0.46$ , Spearman  $\rho = 0.083$ ). Thus, while stabilizing selection on gene expression (as captured by the ECC) can shape the types of mutations that accumulate in the population, it may have little effect on the overall rate at which mutations accumulate in promoter regions within populations, which has been previously used to test for evidence of selection.

### **Stabilizing selection shapes robustness**

While a gene's ECC (computed from the natural genetic variation in regulatory DNA) represents the imprint of its evolutionary history, its mutational robustness (assessed directly from the gene's promoter sequence) should describe how *future* mutations would affect its expression<sup>43</sup>. Across all native yeast promoters, the magnitude of expression changes predicted by the transformer model due to single base-pair mutations follows a power law with an exponent of 2.252 (standard error of fit  $\sigma = \pm 0.002$ ,  $P = 2.4 \times 10^{-263}$ ), such that a small number of mutations have an outsized effect

on expression (~10% of mutations account for ~50% of the changes in expression, **Extended Data Fig. 4d**). In individual genes, the distribution can vary substantially (below).

For a given promoter sequence, we defined the mutational robustness of a sequence length  $L$ , as the percent of its  $3L$  single nucleotide mutational neighbors predicted by the transformer model to result in a *negligible* change in expression (**Extended Data Fig. 4c, Methods**), following previous definitions of mutational robustness<sup>25,43</sup>. The mutational robustness of a gene's promoter sequence was positively correlated with the gene's fitness responsivity (**Supplementary Fig. 2f**, Spearman  $\rho = 0.476$ ,  $P = 8.18 \times 10^{-6}$ ), suggesting that fitness-responsive genes have evolved more mutationally robust regulatory sequences. Mutational robustness, which, unlike the ECC, is computed for single sequences without a set of variants across a population, was also correlated to the ECC (**Supplementary Fig. 2g**, Spearman  $\rho = 0.515$ ,  $P = 9.99 \times 10^{-7}$ ). Similarly, the promoter sequences of yeast genes with conserved expression across *Saccharomyces* strains<sup>41</sup>, Ascomycota species, or mammals<sup>42</sup> had higher mutational robustness ( $P = 8.4 \times 10^{-3}$ ,  $6.5 \times 10^{-5}$ , and 0.00377, respectively, two-sided Wilcoxon rank-sum test).

Thus, genes whose expression levels are under stabilizing selection have regulatory sequences that tend to be more robust to the impact of mutations, which may reflect their history and constrain their future.

### **Fitness landscapes in evolvability space**

Mutational robustness enables the exploration of novel genotypes that could subsequently facilitate adaptation and thus promote evolvability, the ability of a system to generate heritable phenotypic

variation<sup>25</sup>. To characterize regulatory evolvability, we extended our description of mutational robustness by representing each sequence using a sorted vector of expression changes (predicted by the transformer model) that are accessible through single nucleotide mutations (**Fig. 4a**, left, **Methods**). This ‘evolvability vector’ captures the capacity for changes in genotype to alter expression phenotype, in line with previous definitions of evolvability<sup>25</sup>.

We next asked whether regulatory evolvability vectors fell into distinct classes by identifying evolvability ‘archetypes’. Archetypes<sup>44</sup> represent the extremes of canonical patterns, such that the evolvability vector of each individual sequence can be represented by its similarity to each of several archetypes representing these extremes. Applying this paradigm, we used our transformer model to compute evolvability vectors for a new random sample of a million sequences and then learned a two-dimensional representation of these evolvability vectors (referred to as the ‘evolvability space’) using an autoencoder<sup>45</sup> (**Fig. 4a**, right, **Methods**). This archetypal evolvability space, that is bounded by a simplex whose vertices represent evolvability archetypes (**Fig. 4a**, right, **Methods**) and where the evolvability vector of each sequence is a single point, allows us to effectively visualize arbitrarily large sequence spaces in two dimensions.

Three archetypes captured most of the variation in evolvability vectors (**Extended Data Fig. 6a,b**; **Methods**), corresponding to local expression minimum ( $A_{\text{minima}}$ ), local expression maximum ( $A_{\text{maxima}}$ ), and malleable expression ( $A_{\text{malleable}}$ ) (**Fig. 4b**).  $A_{\text{minima}}$  and  $A_{\text{maxima}}$  correspond to sequences where most  $3L$  mutational neighbors do not change expression, and the ones that do, increase it (for  $A_{\text{minima}}$ ) or decrease it (for  $A_{\text{maxima}}$ ). Conversely, for  $A_{\text{malleable}}$  sequences, most  $3L$  mutational neighbors change expression and are equally likely to decrease or increase it (**Fig. 4b**).

In addition to these three archetypes, mutationally robust sequences were present as a central cleft in the evolvability space (**Fig. 4b,c**; “robust”). The evolvability space also distinguishes native regulatory sequences by their associated expression level (**Fig. 4d**), with intermediate expression more likely to be near the malleable archetype ( $A_{\text{malleable}}$ ) and depleted near the robustness cleft (**Fig. 4d, Supplementary Information**).

The location of sequences in evolvability space reflects the selective pressures operating on the sequence. Sequences under strong stabilizing selection on gene expression tend to be located far away from the malleable archetype: there is a strong negative correlation between malleable archetype proximity and mutational robustness (**Extended Data Fig. 6c,e**; Spearman's  $\rho = -0.746$ ,  $P = 1.97 \times 10^{-15}$ ), the ECC (**Extended Data Fig. 6d,f,g**;  $\rho = -0.596$ ,  $P = 5.4 \times 10^{-9}$ ), fitness responsivity (**Extended Data Fig. 6h**;  $\rho = -0.413$ ,  $P = 1.4 \times 10^{-4}$ ), and expression conservation across species as measured by RNA-seq (*Saccharomyces*:  $P = 0.000251$ , Ascomycota:  $P = 0.00002$ , Mammals:  $P = 0.00114$ ; two-sided Wilcoxon rank-sum test).

To visualize promoter fitness landscapes in two dimensions we combined our sequence-to-expression transformer model with previously measured expression-to-fitness curves<sup>11</sup>, and integrated them with the two-dimensional archetypal evolvability space (**Fig. 4e, Extended Data Fig. 7, Methods**). Unlike prior visualizations of fitness landscapes, which group sequences by their sequence similarity, here, sequences are arranged by the similarity in their evolvability. This approach effectively visualizes arbitrarily large sequence spaces in two-dimensions, as well as groups sequences by their evolutionary properties. This addresses the challenges otherwise posed by sequence similarity-based landscapes since highly similar regulatory sequences can have

different functional properties (*e.g.*, due to a loss of a TF binding site), while very different sequences can be functionally similar (*e.g.*, due to shared TF binding sites). When organismal fitness is available for a particular gene and overlaid on the landscape (**Fig. 4e**, **Extended Data Fig. 7**), the resulting patterns depend on both the condition-specific sequence-to-expression function (*e.g.*, governing color (fitness) through predicted expression, and embedded position, through evolvability) and the gene- and condition-specific expression-to-fitness functions.

Finally, we studied how natural yeast sequences explored evolutionary space, by placing the evolvability vectors of each of set of orthologous promoters of the 1,011 sequenced *S. cerevisiae* isolates<sup>37</sup> in the archetypal evolvability space. When a gene's promoter from one strain is near the malleable archetype, its orthologs in the other strains tended to broadly distribute in the evolvability space (**Extended Data Fig. 6i**), but avoid the robustness cleft (*e.g.*, the *DBP7* promoter from strain S288C; **Fig. 4f**). Conversely, when a promoter is near the robustness cleft (*e.g.*, the *UTH1* promoter from S288C), so are its orthologs (**Fig. 4g**, **Extended Data Fig. 6i**). Using *in silico* mutagenesis to interpret our model, we found that the *DBP7* promoter is particularly malleable partly as a result of an intermediate affinity Rap1p binding site, where the most impactful mutations increased or decreased the Rap1p affinity for this site, impacting expression (**Extended Data Fig. 8a**). By contrast, the *UTH1* promoter requires many sequential mutations, each of which has minimal impact individually, to reduce expression appreciably (**Extended Data Fig. 8b**). This could reflect the ways in which stabilizing selection constrains evolvability: promoters that are not under strong stabilizing selection explore expression space more freely and can quickly adapt to a new expression optimum, since the population likely already contains multiple alleles that achieve

diverse expression levels (*e.g.* **Fig. 4f**). Interestingly, many of the native sequences in *S. cerevisiae* are near the robustness cleft (**Fig. 4h**).

Thus, the evolvability vector, which can be computed using our model directly for any sequence (without any population genetics data), encodes information about a sequence's evolutionary history and potential futures.

## DISCUSSION

Here, we presented a framework that addresses fundamental questions in the evolution and evolvability of regulatory sequences<sup>2,25</sup>. Our models, developed using a combination of large scale random sequence libraries, sensitive reporter assays and deep learning (**Methods**), are useful as “oracles” for model-guided biological sequence design<sup>19</sup>, and answering important questions in the study of fitness landscapes<sup>4-6</sup>, evolutionary malleability of expression and its variation across strains and species<sup>2</sup>, mutational robustness<sup>43</sup>, and evolvability<sup>25</sup>. The framework presented here will help advance synthetic biology, cell and gene therapy, and metabolic engineering in addition to the study of evolution.

Previous studies suggested that evolution favors more complex regulatory solutions<sup>46</sup>, but we showed that if stabilizing selection acts only on expression, regulatory complexity extremes gradually move towards the moderate complexity levels observed in native and random sequences (**Fig. 2d**). This supports a model where most extant regulatory sequences evolved by sampling constraint-satisfying solutions in proportion to their frequency in the sequence space, without specific consideration of the solution's complexity.



In our study, evolving condition-specificity in a promoter sequence was much slower than simply modifying the expression level. Some yeast genes achieve condition-specificity by including multiple binding sites for condition-responsive TFs. For instance, the *GALI-10* Upstream Activating Sequence contains multiple binding sites for the galactose-responsive Gal4, which are conserved across millions of years, suggesting an ancient origin<sup>47</sup>. Because the size of the regulatory region restricts the number of TF binding site locations, including more TFs and more regulatory sequences per gene (*e.g.* enhancers) may be required for more complex regulatory programs observed in higher eukaryotes<sup>48</sup>.

The  $d_N/d_S$  ratio has been used extensively to characterize the evolutionary rates of protein coding genes<sup>34</sup>, and we developed an analogous<sup>2,35</sup> coefficient, the ECC, for detecting evidence of selection on expression from natural variation across multiple orthologous regulatory sequences in strains of one species. The ECC complements and extends existing measures of expression conservation, since it integrates across the regulatory sequence and is not limited to specific TFs or binding motifs, does not require additional experiments to test the functions of mutations for each regulatory region, and does not rely on detecting non-uniformity in mutation distributions.

Complementing the ECC, mutational robustness as calculated with our model is predictive of selective pressures on individual sequences (**Supplementary Fig. 2f-g**). While we find that strong constraint on the function of regulatory sequences can shape them to be robust to future mutations, it is unlikely that robustness itself is the selected trait, since increased robustness to future mutations is likely to be of little marginal benefit<sup>43</sup>. Instead, this may reflect a secondary benefit

of having evolved decreased expression noise<sup>49,50</sup>, or another as-yet-unknown mechanism. It may also reflect the fact that the sequences of some ancestral promoters may be similar to the mutational neighbors of extant sequences, and, if selective constraints on expression have remained stable, these ancestral and extant sequences likely have similar expression levels.

Based on our model-derived evolvability vectors, sequences spanned an evolvability spectrum from robust to malleable (**Fig. 4c-d,f-h**), and for native regulatory sequences, the magnitudes of accessible mutation effects follows a power law. Evolvability vectors also help visualize fitness landscapes<sup>4</sup> (**Fig. 4e, Extended Data Fig. 7**) and future work can further improve our understanding of their topography<sup>4,5</sup>.

Our sequence-to-expression models are currently limited by regulatory region and species. For example, sequence mutations that affect other regulatory mechanisms (*e.g.*, genomic context, mRNA processing and degradation, regulation by RNA-binding proteins, translational efficiency) can compensate for those that affect transcription. While our models emulated the biological process of our experimental system, as demonstrated by their excellent predictive power, future interpretability studies will shed further light on molecular mechanisms. Finally, for multicellular organisms, selection acts simultaneously on expression levels in many different cell types and environments. As models of gene regulation are created for other species, environments, and regulatory regions, our framework will help provide further insights into regulatory evolution.

## **Acknowledgements**

We thank Google TPU Research Cloud for TPU access, Leslie Gaffney for help with figure preparation, Broad Genomics Platform for sequencing work, Jan-Christian Hütter for advice on fitness responsiveness, Jenna Pfiffner-Borges for help with RNA-seq, Ruby Yu, Byron Lee and Nima Jaberri for manuscript feedback and members of the Regev lab for discussions. E.D.V. was supported by the MIT Presidential Fellowship. C.G.D. was supported by a Canadian Institutes for Health Research Fellowship and the NIH (K99-HG009920-01). F.A.C. and J.M. were supported by ANID - Programa Iniciativa Científica Milenio - ICN17\_022. Work was supported by the Klarman Cell Observatory and HHMI. A.R. was an Investigator of the HHMI.

## **Conflict of Interest statement**

A.R. is a co-founder and equity holder of Celsius Therapeutics, an equity holder in Immunitas, and until July 31, 2020 was an S.A.B. member of ThermoFisher Scientific, Syros Pharmaceuticals, Neogene Therapeutics and Asimov. From August 1, 2020, A.R. is an employee of Genentech. The other authors declare no competing interests.

## **Data Availability**

Data generated for this study are available on NCBI's GEO, accession numbers GSE163045 and GSE163866. All models and processed data are available on Zenodo at <https://zenodo.org/record/4436477>.

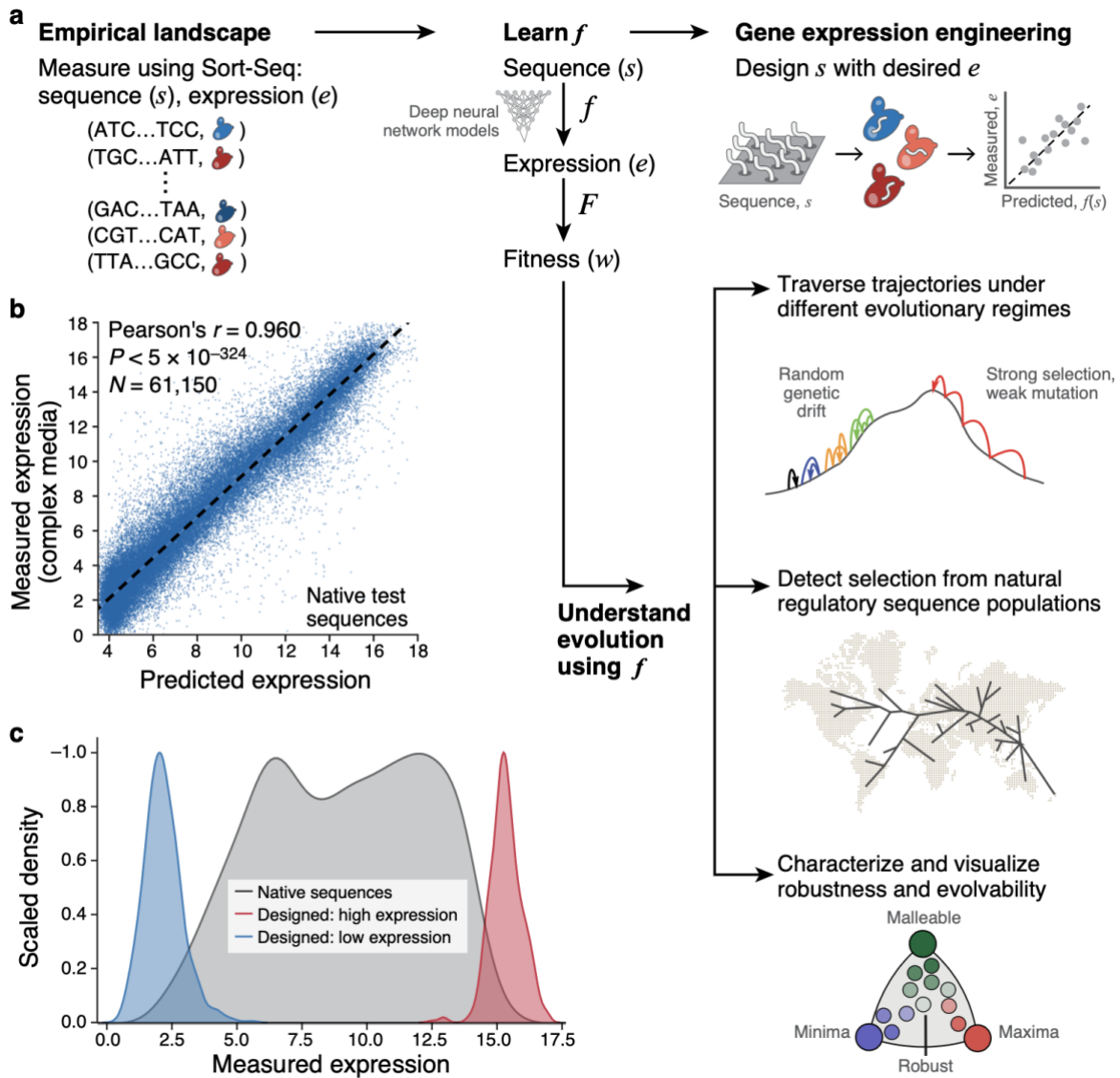
## **Code Availability**

Code is available on GitHub at <https://github.com/ledv/evolution> and CodeOcean at <https://codeocean.com/capsule/8020974/tree>. A web app is available at <https://ledv.github.io/evolution/>.

### **Author contributions**

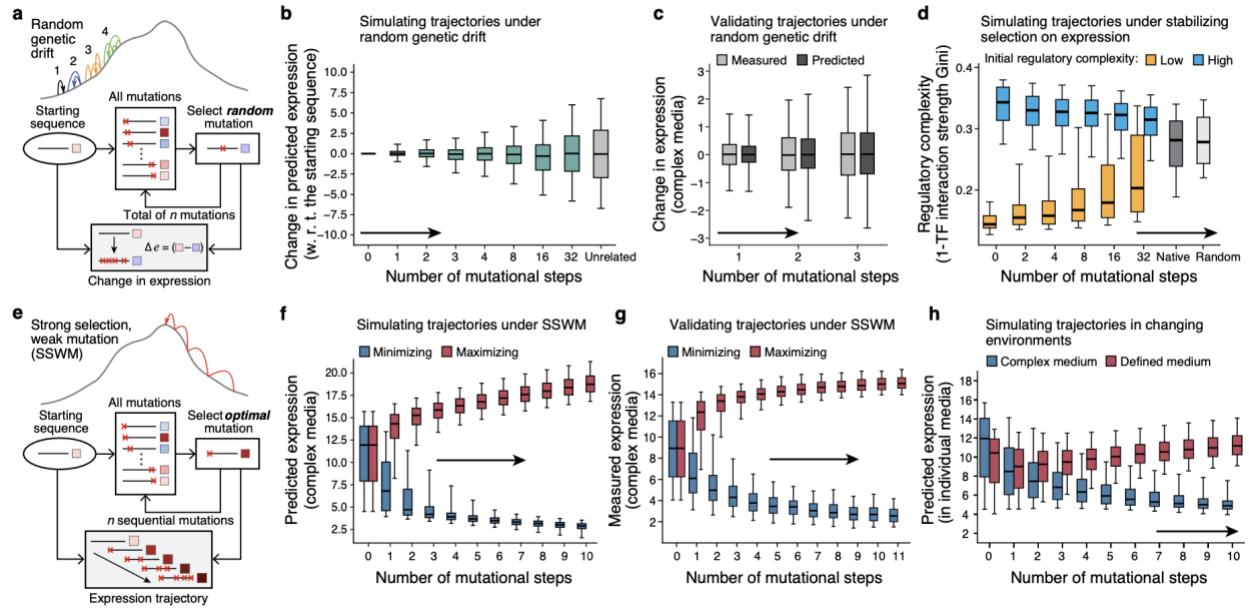
E.D.V., C.G.D. and A.R. conceived, designed and supervised the study. E.D.V. and C.G.D. carried out the analyses. M.Y., L.F., X.A. and D.A.T. performed and J.Z.L. supervised the Ascomycota cross-species RNA-seq experiments. J.M. performed and F.A.C. supervised the *CDC36* experiments. E.D.V. and C.G.D. performed the rest of the experiments. E.D.V., C.G.D. and A.R. wrote the manuscript.

## FIGURES



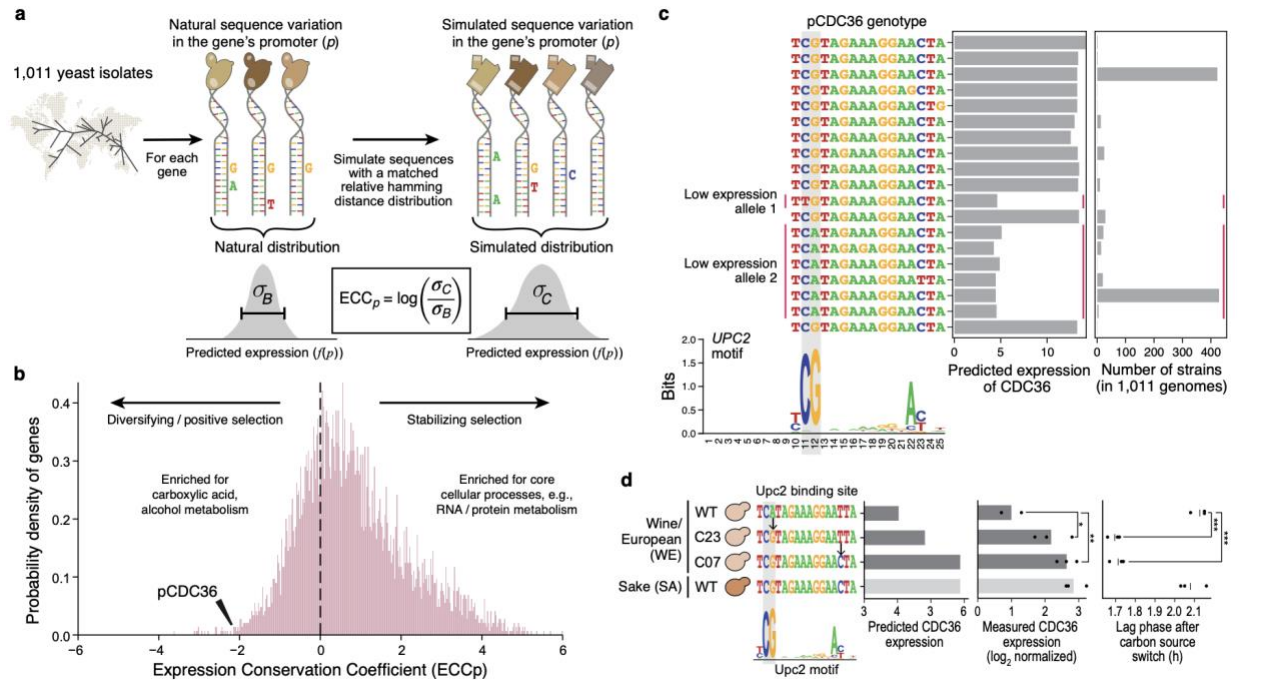
**Fig. 1 | The evolution, evolvability, and engineering of gene regulatory DNA.**

**a**, Project overview. **b**, Prediction of expression from sequence using the model. **b**, Predicted ( $x$  axis) and experimentally measured ( $y$  axis) expression in complex media (YPD) for native yeast promoter sequences. Pearson's  $r$  and associated two-tailed  $p$ -values are shown; dashed line: line of best fit. **c**, Engineering extreme expression values beyond the range of native sequences using a genetic algorithm (GA) and the sequence-to-expression model. Normalized kernel density estimates of the distributions of measured expression levels for native yeast promoter sequences (grey), and sequences designed (by the GA) to have high (red) or low (blue) expression.



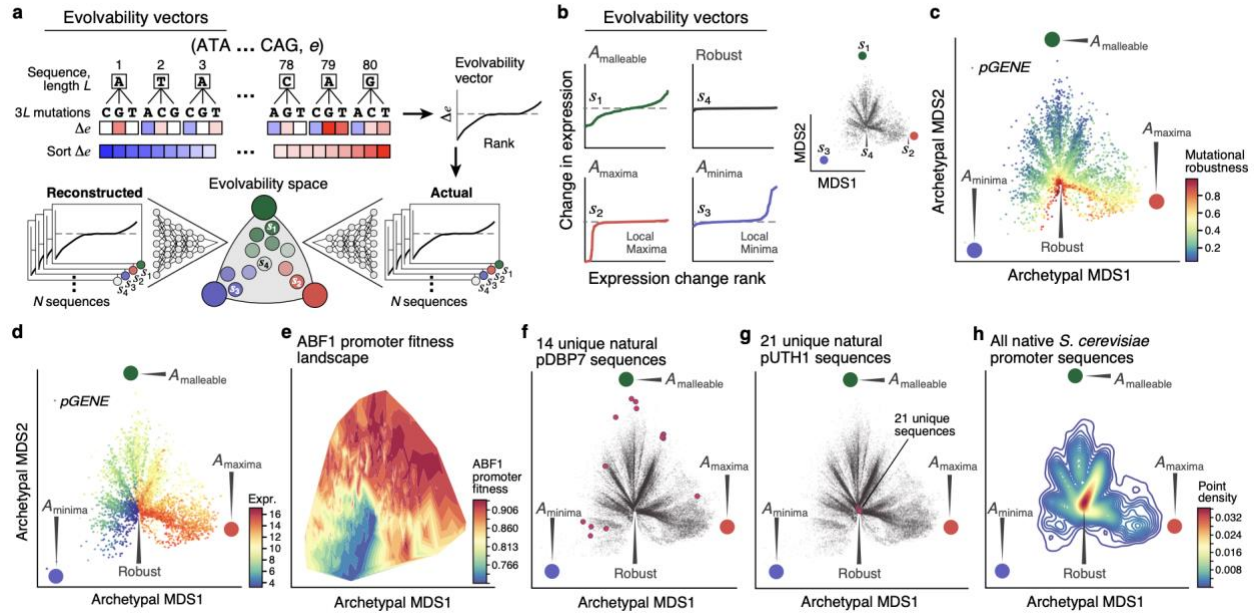
**Fig. 2 | The evolutionary malleability of gene expression.**

**a-c**, Expression divergence under genetic drift. **a**, Simulation procedure. **b**, Predicted expression divergence. Distribution of the change in predicted expression ( $y$  axis) for random starting sequences ( $n=5,720$ ) at each mutational step ( $x$  axis) for simulated trajectories. Silver bar: expression differences between unrelated sequences. **c**, Experimental validation. Distribution of measured (light grey) and predicted (dark grey) changes in expression in complex media ( $y$  axis) for synthesized randomly-designed sequences ( $n=2,983$ ) at each mutational step ( $x$  axis). **d**, Stabilizing selection on gene expression leads to moderation of regulatory complexity extremes. Regulatory complexity ( $y$  axis) of sequences from sequential mutational steps ( $x$  axis) under stabilizing selection to maintain the starting expression levels, where the regulatory interactions of starting sequences are complex (blue;  $n=192$ ) or simple (orange,  $n=172$ ). Right bars: regulatory complexity for native (dark grey) and random (light grey) sequences. **e-g**, Sequences under strong-selection weak-mutation (SSWM) can rapidly evolve to expression optima. **e**, Simulation procedure. **f**, Predicted expression evolution. Distribution of predicted expression levels ( $y$  axis) in complex media at each mutational step ( $x$  axis) for trajectories favoring high (red) or low (blue) expression, starting with native promoter sequences ( $n=5,720$ ). **g**, Experimental validation. Measured expression distribution in complex media ( $y$  axis) for the synthesized sequences ( $n=10,322$  sequences; 877 trajectories) at each mutational step ( $x$  axis), favoring high (red) or low (blue) expression. Axis scales differ due to variation in measurement procedure (**Supplementary Information**). **h**, Competing expression objectives constrain expression adaptation. Distribution of predicted expression ( $y$  axis) in complex (blue) and defined (red) media at each mutational step ( $x$  axis) for a starting set of native promoter sequences ( $n=5,720$ ) optimizing for high expression in defined (red) and simultaneous low expression in complex (blue) media. (**b-d,f-h**) Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range.



**Fig. 3 | The Expression Conservation Coefficient (ECC) detects signatures of stabilizing selection on gene expression using natural genetic variation in regulatory DNA. a,** ECC calculation from 1,011 *S. cerevisiae* genomes<sup>37</sup>. **b,** ECC distribution for *S. cerevisiae* genes.

Frequency distribution of ECC values ( $x$  axis). Dashed line separates regions corresponding to disruptive/positive selection (left) and stabilizing selection (right). GO terms enriched by the ECC ranking are shown. Arrowhead: ECC value for the *CDC36* promoter sequence. **c,** Convergent regulatory evolution in the *CDC36* promoter. Predicted expression ( $x$  axis, left bar plot) and associated number of strains ( $x$  axis, right bar plot) of all alleles among the analyzed *CDC36* promoter sequence within 1,011 yeast isolates, along with an alignment of their Upc2p binding site sequences (left; Upc2p binding motif below). Red vertical lines: two independently evolved low-expressing alleles. Grey vertical boxes: key positions in the Upc2p motif with single nucleotide polymorphisms. **d,** Validation of *CDC36* promoter allele expression and organismal phenotype. Strains ( $y$  axis) with different Upc2p binding site alleles for both model-predicted *CDC36* expression (left; predicted on -170:-90 region to capture entire Upc2p binding site), measured *CDC36* expression (middle), and lag phase duration (right). Points: biological replicates ( $n=3$ ); bars/vertical lines: means. Bar color: strain background. Student's t-test p-values, unpaired, equal variance, one-sided (expression, WE WT vs. C23  $p=0.044$ , C07  $p=6.69 \times 10^{-3}$ ) or two-sided (lag phase, WE WT vs. C23  $p=1.34 \times 10^{-4}$ , C07  $p=2 \times 10^{-4}$ ); \* $p<0.05$ ; \*\* $p<0.01$ ; \*\*\* $p<0.001$ .



**Fig. 4 | The evolvability vector captures fitness landscapes.**

**a**, Characterizing regulatory evolvability by computing an evolvability vector. Left and middle: Generating evolvability vectors for a sequence. Right: training an autoencoder with evolvability vectors to generate a 2D representation to visualize sequences in archetypal evolvability space.

**b**, Evolvability archetypes discovered by the autoencoder. Left: Evolvability vectors of the rank ordered ( $x$  axis) predicted change in expression ( $y$  axis) for native sequences closest to each of the malleable (green), maxima (red) or minima (blue) archetypes and the ‘robustness cleft’ (black). Right: all native yeast (*S. cerevisiae* S288C) promoter sequences (grey points) projected onto the archetypal evolvability space by their evolvability vectors. Evolvability archetypes (colored circles) and their closest native sequences ( $s_1$ - $s_4$  as on left) are marked.

**c,d**, Evolvability space captures mutational robustness and expression levels. Evolvability vectors (points) of all native yeast promoter sequences projected onto the evolvability space (archetypes are large colored circles, as in **b**) and colored by mutational robustness (**c**) or predicted expression levels (**d**).

**e**, *ABF1* promoter fitness landscape. Evolvability vectors of promoter sequences projected onto the evolvability space and colored by computed fitness (color, **Methods**).

**f,g**, Malleable promoter sequences dynamically traverse the evolvability space. Evolvability vector projections of native sequences (points) from all 1,011 *S. cerevisiae* isolates. Red points: natural promoter sequence variants for *DBP7*, the promoter closest to the malleable archetype (**f**) and for *UTH1*, the promoter closest to the robustness cleft (**g**).

**h**, The robustness of native promoter sequences. Density (color) of all native yeast promoter sequences when their evolvability vectors are projected onto the evolvability space.



## METHODS

### Experimental measurement of sequence-expression pairs using a Sort-seq strategy

We experimentally measured expression using a Sort-seq<sup>2,3,51-59</sup> strategy called the Gigantic Parallel Reporter Assay (GPRA) we previously described<sup>26</sup> (**Supplementary Fig. 1**). Briefly, for each set of expression measurements mentioned, random or designed single stranded oligonucleotides were ordered from IDT (random; **Supplementary Table 3**) or Twist Biosciences (designed; sequences on GEO; accession GSE163045), cloned into the promoter of a Yellow Fluorescent Protein (YFP) gene within a CEN plasmid (Addgene: 127546) as previously described<sup>26</sup> and transformed into yeast (strain Y8205 for the training dataset of random sequences, and strain S288C::*ura3* for all the rest of the sequences measured). The library is maintained in yeast as an episomal low copy number plasmid. It was previously reported that the expression measurements are highly correlated with expression levels as measured using integrated reporters ( $R^2=0.97$ )<sup>54</sup>. Yeast were grown in continuous log phase, diluting as necessary to maintain an OD between 0.05 and 0.6 for 8-10 generations up until the time of harvest. Cells were harvested, washed once in ice cold PBS, and kept on ice in PBS until sorting. Cells were sorted into 18 uniformly-sized expression bins covering the majority of the expression distribution. Post sort, cells were re-grown in SD-Ura until saturation, plasmids isolated, and sequencing libraries created sequenced with a 150 cycle NextSeq kit. For libraries with random 80 bp sequences, sequences were consolidated as previously described<sup>26</sup>. Reads from other (defined, non-random) libraries were aligned to the pre-defined sequences using Bowtie2<sup>60</sup>, including only reads that perfectly matched a designed sequence. For each sequence, the expression level was the average of the expression bins in which it was observed, weighted by the number of times it was observed in each bin. These expression measurements were carried out separately in defined media lacking uracil

(SD-Ura (Sunrise Science, #1703-500)) and complex media (YPD: yeast extract, peptone, dextrose).

### **Architecture of the convolutional model**

A deep neural network model<sup>20,21,23,61-69</sup> with convolutional layers was constructed and used for designing sequences with high and low expression (**Fig. 1c**), and running evolutionary simulations under stabilizing selection, genetic drift, and SSWM (**Fig. 2**) for each condition. These designed sequences, whose expression was experimentally quantified (e.g. **Fig. 1c** and **2d,g**), were designed using models with the following architecture:

**Input.** The input is the DNA sequence ( $s$ ) represented in one-hot encoding. Input Shape: (1, 110, 4)

#### ***Convolution Block***

- For the forward and reverse strand, separately,
  - o Strand-specific convolution layer 1. Kernel Shape: (1, 30, 4, 256)
  - o Strand-specific convolution layer 2. Kernel Shape: (30, 1, 256, 256)
- Concatenation of features from the forward and reverse strand
- Convolution layer 3. Kernel Shape: (30, 1, 512, 256)
- Convolution layer 4. Kernel Shape: (30, 1, 256, 256)
- A bias term and a ReLU activation was added to each convolution layer in this block.

#### ***Fully Connected Layers***

- Fully connected layer 1. Kernel Shape: (110\*256, 256).

- Fully connected layer 2. Kernel Shape: (256, 256)
- A bias term and a ReLU activation were added to each layer in this block.

**Output.** Linear combination of the 256 features extracted as a result of all the previous operations on the sequence ( $s$ ) to generate the predicted expression ( $e$ ).

Every fully connected layer was  $L2$  regularized with a 0.0001 weight and had a dropout probability of 0.2.

### **Training of the convolutional model**

For training, 20,616,659 random sequences for the defined medium and 30,722,376 random sequences for the complex medium (each to train a separate model) were used, along with their experimentally measured expression as described above. A mini-batch size of 1,024 was used for training and a mean squared error loss was optimized using the Adam optimizer<sup>70</sup> with an initial learning rate of 0.0005. The model was trained for 5 epochs. Model architecture was written in TensorFlow<sup>71</sup> 1.14 using Python 3.6.7. The convolutional model used TensorFlow graphs and sessions in its implementation and was thus incompatible with the Tensor Processing Units (TPUs)<sup>72</sup>. These convolutional models (for both media) were used for all the predictions in **Fig. 1** and **2** and **Extended Data Fig. 1, 2**.

The models were tested by predicting expression on sequences that the model had never seen before (**Supplementary Fig. 21**) that were measured in separate experiments, where the library was lower complexity (fewer sequences) than the experiments that generated the training data,

such that the expression associated with each sequence was measured with high accuracy (~100 yeast cells per sequence on average). The test libraries included random, native (*i.e.* present in the yeast genome), and designed sequences.

Training and evaluation were carried out on 4 Tesla M60 GPUs. All code for training and using the convolutional model is available here:

[https://github.com/ledv/evolution/tree/master/manuscript\\_code/model/gpu\\_only\\_model](https://github.com/ledv/evolution/tree/master/manuscript_code/model/gpu_only_model).

### **Architecture of the transformer model**

A transformer model<sup>23,32,73</sup> was developed to run inference faster than the convolutional model, as needed for the evolutionary analyses in **Fig. 3 and 4**. The transformer model had ~20x fewer parameters (~1.3 million, compared to the ~24 million parameters of the convolutional model) and was able to leverage Tensor Processing Units (TPUs) for computation. Transformer models are used in all the analyses in **Fig. 3 and 4** and **Extended Data Fig. 3-4, and 6-8**. Benchmarking analyses and ablation analyses for the transformer model are available in the **Supplementary Information**.

The deep transformer model has the following architecture (**Supplementary Fig. 12**):

**Input.** The input is the DNA sequence ( $s$ ) represented in one-hot encoding. Input Shape: (110, 4)

**Convolution Block.** The convolution block is constructed in the following order (**Supplementary Fig. 12b**):

- Reverse Complement Aware 1D Convolution. The forward and reverse strand are operated on separately with a convolutional kernel to generate strand specific sequence-environment interaction features. Kernel Shape: (30, 4, 256).
- Batch Normalization
- Rectified Linear Unit (ReLU)
- Concatenation of Features from the forward and reverse strand
- 2D Convolution: Convolve over the combined features from both the strands to capture interactions between strands. Kernel Shape: (2, 30, 4, 256)
- Batch Normalization
- ReLU
- 1D Convolution. Kernel Shape: (30, 64, 64)
- Batch Normalization
- ReLU

**Transformer Encoder Blocks.** Two transformer encoder blocks<sup>32,74,75</sup> are constructed in the following order (**Supplementary Fig. 12c**):

- Multi-Head Attention: 8 heads, capturing relations between features from different positions of (*s*) to compute a representation for the features extracted from the convolution block from (*s*).
- Residual Connection
- Layer Normalization
- Feed Forward Layer with 8 units
- Residual connection
- Layer Normalization

***Bidirectional LSTM layer.*** A bidirectional LSTM layer to capture the long-range interactions between different regions of the sequence with 8 units and 0.05 dropout probability.

***Fully Connected Layers (Supplementary Fig. 12d).*** Two Fully connected layers with 64 Hidden Units, each consisting of ReLU and Dropout (0.05 dropout probability).

***Output.*** Linear Combination of 64 features extracted as a result of all the previous operations on the sequence ( $s$ ) to generate the predicted expression ( $e$ ).

### **Training of the transformer model**

20,616,659 random sequences (defined medium) and 30,722,376 random sequences (complex medium), along with their experimentally measured expression, were used to train separate models for each media. Model architecture was written in TensorFlow<sup>71</sup> 1.14 using Python 3.6.7 with multiple open source libraries (citations, where relevant, are included in code for them). A mini-batch size of 1,024 was used for training and a mean squared error loss was optimized using a RMSProp optimizer<sup>76</sup> with a learning rate of 0.001. The stopping criterion monitored was the ‘r-squared’ value and the model was allowed to train for 10 epochs without improvement before stopping training. Training was carried out on a Google Cloud Tensor Processing Unit (TPU)<sup>72</sup> v3-8. Evaluation was carried out on 4 Tesla M60 GPUs. The model architecture visualization was generated using Netron 4.5.1. All processed data and models are publicly available on Zenodo at <https://zenodo.org/record/4436477> and all code is available on GitHub at [https://github.com/ledv/evolution/tree/master/manuscript\\_code/model/tpu\\_model](https://github.com/ledv/evolution/tree/master/manuscript_code/model/tpu_model). Transformer models are used in all the analyses in **Fig. 3** and **4** and **Extended Data Fig. 3-4, and 6-8**.

The models were tested by predicting expression on test sequences that the model had never seen before (**Supplementary Fig. 21**) that were measured in separate experiments, which included random, native (*i.e.* present in the yeast genome), and designed sequences. To obtain expression measurements for each tested sequence that are more accurate than those from the high-complexity training data experiment, library complexity was limited such that each test promoter sequence is observed in ~100 yeast cells (**Methods, Supplementary Information**).

### **Gene expression engineering using a genetic algorithm for sequence design**

To design<sup>77–80</sup> new sequences with desired expression, a genetic algorithm (GA) was implemented with the distributed evolutionary algorithms in python (DEAP) package<sup>81</sup>. The mutation probability and the two-point crossover probability were set to 0.1 and the selection tournament size was 3. The initial population size was 100,000 and the GA was run for 10 generations. The convolutional model was used as the basis for the objective function for GA, which was maximized for high expression and minimized for low expression (maximizing negative predicted expression). The top 500 sequences were synthesized (by Twist Biosciences) and expression was measured experimentally using our reporter assay, as described above.

### **Characterizing random genetic drift**

Simulation of random genetic drift (**Fig. 2a**) was initialized with a set of 5,720 random sequences, in generation 0. For each sequence in this starting set, a new single sequence was randomly picked from its  $3L$  mutational neighborhood (the set of all sequences at a Hamming distance of 1 from a sequence of length  $L$ ) and the difference in expression between the new sequence and the starting

sequence was calculated using the convolutional model (**Fig. 2b**). This was done for each starting sequence to get generation 1. Each subsequent generation  $n$  was produced by picking a single sequence randomly from the  $3L$  mutational neighborhood of each sequence in the preceding generation  $n-1$ . The simulation was carried out for 40 generations. Simulations were also subsequently repeated with the transformer model (**Extended Data Fig. 3f**), yielding concordant results.

For experimental validation, 1,000 random starting sequences were synthesized, introducing between one to three random mutations to these sequences. The expression levels of starting and mutated sequences were measured in both complex and defined media experimentally using our reporter assay. For 990 of these 1,000 starting sequences, experimental measurements were available for all three mutational distances. Additionally, 20 (median) separate single mutations were introduced to each of 196 native sequences, the sequences were synthesized, and their associated expression was measured similarly for both of these media; these were also included in the boxes for one mutational step in **Fig. 2c** and **Extended Data Fig. 1e**.

### **Characterizing the regulatory complexity of a sequence**

To estimate the regulatory complexity<sup>82,83</sup> of a sequence, the Gini coefficient of the regulatory interaction strengths for each TF was calculated. A new biochemical model was first trained with our defined media data to complement the existing one trained on complex media, using our published model architecture of TF binding and position-aware activity<sup>26</sup> and the training procedure previously described<sup>26</sup> (**Supplementary Notes**). The regulatory interaction strength was



then individually calculated for each regulator by setting the concentration parameter for that TF (individually) to 0 in the learned model, and the biochemical model was used to quantify the resulting change in expression, as previously described<sup>26</sup>. The resulting vector of interaction strengths was used to calculate a Gini coefficient for each sequence, separately for the complex and defined media models. The Gini coefficient is a measure of inequality of continuous values within a population, most commonly applied to wealth or income, and ranges from 0 (all members of the population have equal wealth) to 1 (the wealth of a population is held by a single individual). Regulatory complexity for a sequence is then 1-Gini, such that 1 indicates that all TFs contribute equally to the regulation of the gene and 0 indicates that a single TF is solely responsible for its regulation. As starting points for our trajectories, 200 native promoter sequences (from -160 to -80, relative to the TSS) were chosen with relatively high regulatory complexity and another 200 were chosen with relatively low regulatory complexity, spanning the range of predicted expression levels, as starting points for our trajectories.

Trajectories for stabilizing selection on gene expression were designed using the convolutional model (**Fig. 2d**). Here, all sequences were required to maintain a predicted expression level within 0.5 of the original expression levels at all steps along the trajectory. There was no explicit constraint on regulatory complexity in this simulation of stabilizing selection. In order to ensure that expression was unchanged, expression levels were measured experimentally for sequences along a trajectory at growing mutational steps from the initial sequence (2, 4, 8, 16, 32 mutations). Any trajectories for which an expression measurement was missing for *any* experimentally tested sequence were excluded from all analyses, retaining 172 trajectories with initial low regulatory complexity and 192 trajectories with initial high regulatory complexity. Testing whether observed

trends in regulatory complexity were affected by the degree to which expression was either predicted (by the convolutional model for 1-32 mutations) or observed (by the experiment at 2, 4, 8, 16, or 32 mutations) to be conserved, showed that the trends were robust to the degree of expression conservation (**Supplementary Fig. 11**).

### **Characterizing directional trajectories under SSWM**

Simulations of trajectories under a Strong Selection-Weak Mutation (SSWM)<sup>84-86</sup> regime were initialized with a set of native yeast promoter sequences (defined here as the subset from -160 to -80 relative to the TSS for all the genes in the yeast reference genome for which we had a good TSS estimate (Supplementary Table 3 in <sup>26</sup>) as the starting generation 0. For each sequence in generation  $n$ , the sequence from its  $3L$  mutational neighborhood that had the maximal (or separately, minimal) predicted expression using the convolutional model was picked to get generation  $n+1$ . The simulation was carried out for 10 rounds separately in the complex (**Fig. 2f**) and defined (**Extended Data Fig. 1f**) media. The simulations were subsequently repeated using the transformer model (**Extended Data Fig. 3i-j**).

For experimental validation, a subset of sequences from several generations were synthesized along mutational trajectories simulated by the convolutional model for complex media (10,322 sequences from 877 trajectories, 805 of which had every sequence along the trajectory successfully measured) and one for defined media (6,304 sequences from 637 trajectories, 591 of which had every sequence along the trajectory successfully measured) and their expression was measured in

the corresponding media experimentally using our reporter assay (**Fig. 2g, Extended Data Fig. 1g**).

### **Measuring the *URA3* expression-to-fitness relationship**

Two complementary environments were studied with opposite selective pressures on the expression of *URA3* (encoding an enzyme responsible for uracil synthesis): defined media, where organismal fitness increases with gene expression (up to saturation) and complex media + 5-FOA, where fitness decreases with Ura3p expression.

Convolutional models trained on defined and complex media were used to choose a set of 11 sequences that span a broad range of predicted expression levels in the two media when cloned into a YFP expression vector<sup>26</sup>. The relationship between expression of *URA3* and organismal fitness in yeast was estimated from experimental measurements with these 11 sequences, by cloning promoter sequence in front of YFP to measure expression level and in front of *URA3* to measure fitness. Unless otherwise noted, yeast were grown at 30°C, in an orbital shaker incubator at 225 RPM. Each vector was transformed into yeast (S288C::*ura3*), and three independent transformants were selected per vector to serve as biological replicates. For measuring expression, yeast were grown overnight in either YPD+NAT (yeast extract, peptone, dextrose, with 75µg/ml nourseothricin) or SD-Ura (synthetic defined media, lacking uracil; Sunrise Science 1703-500), and then re-inoculated in the morning and allowed to grow for 6 hours prior to measuring expression by flow cytometry for each replicate as the log ratio of YFP to the constant background RFP, including only cells obtaining the top 50% of RFP expression. Fitness was obtained by measuring the growth rate of each yeast strain in either SD-Ura or YPD+NAT+5-FOA (0.25 mg/ml 5-FOA). Yeast were grown continuously in triplicate in log phase, with linear shaking at 30°C in

a Synergy H1 plate reader (Biotek), by diluting each well to maintain  $OD < 0.7$ , with OD measured at 15 minute intervals. Growth rate was defined for each replicate as the median of the instantaneous smoothed growth rates over 5 measurements in log phase, considering only time points where  $0.05 < OD < 0.5$ . Each promoter's expression and growth rate were summarized as the mean of the three replicates.

### **Characterizing trajectories under conflicting expression objectives in different environments**

Simulations of sequence evolution in two complementary environments with opposite selective pressures (defined media and complex media) were initialized with a set of native yeast promoter sequences (present at -160 to -80 relative to the TSS) as the starting generation 0, with the objective function defined as the difference in predicted expression between defined and complex media (**Fig. 2h, Extended Data Fig. 2d-g**) using convolutional models trained in the respective media. The difference in expression between the two conditions was maximized at each iteration, which assumes that the cells are exposed to both environments before the mutations can reach fixation, an example of evolution in rapidly fluctuating environments<sup>31</sup>. For simplicity, it is assumed that fitness is directly proportional to higher expression in one condition and to lower expression in the other, such that mutations will be considered favorable even if they decrease fitness in one condition so long as they increase it in the other condition by a greater amount.

One simulation aimed to maximize the expression difference (defined minus complex), and the other to minimize it (maximizing complex minus defined). For each sequence in generation  $n$ , the sequence from its  $3L$  mutational neighborhood that had the maximum (or separately, minimum) value for the objective function based on the convolutional model prediction is picked for

generation  $n+1$ , to a total of 10 generations. The simulations were subsequently repeated using the transformer model yielding similar results (**Supplementary Fig. 17b-f**).

Motifs that were enriched in the sequences of generation 10 compared to the starting sequences were identified *de novo* using DREME<sup>87</sup>, and each of the top 5 consensus motifs were used as queries to search the YeTFaSCo database<sup>88</sup>, reporting the closest match, or one of multiple similar matches.

### **Finding orthologous promoters in the 1,011 *S. cerevisiae* genomes dataset**

To identify orthologs of S288C promoters in the whole genome sequences of the 1,011 yeast strains<sup>37</sup>, BLAT<sup>89</sup> was used to identify regions of  $\geq 80\%$  identity with each -160 to -80 region (relative to the TSS) annotated in the reference S288C genome sequence (R64)<sup>90</sup>. Any strains with more than one such match, where the match contained insertions or deletions, or had incomplete matches, were excluded on a gene-by-gene basis. Genes with more than 1.2 matches with  $\geq 80\%$  identity per genome, on average, were excluded altogether.

### **Computing the expression conservation coefficient (ECC)**

To calculate the ECC (a regulatory analog<sup>2,35,91,92</sup> of  $d_N/d_S$ <sup>34,93,94</sup>), for each yeast gene promoter, the transformer model was used to predict an expression value for each orthologous promoter in the 1,011 yeast genomes (above), defining an expression distribution with a standard deviation  $\sigma_B$ . Next, a set of sequences with random mutations was generated from each gene's consensus promoter sequence (defined as the most abundant base at each position across the strains), such

that the number of sequences at each Hamming distance from the consensus promoter sequence was the same for the natural and simulated sets. Here, mutations introduced to create random variation sampled each base with equal probability; using observed mutation rates yielded similar results (**Supplementary Information**). The same transformer model to predict the expression of the simulated sequences, and calculate its standard deviation  $\sigma_C$ . The nominal ECC is  $\log(\sigma_C/\sigma_B)$ . Because the variance on simulated sequences is better estimated than in natural orthologs (whose sequences may be more constrained), a constant correction factor is subtracted, calculated by creating a second simulated set of randomly mutated sequences whose diversity is limited to the same extent as in the natural set, by creating only one random mutation for every unique sequence in the set of native orthologs. Finally, the expression for this second set of sequences is predicted by the transformer model, and its standard deviation ( $\sigma_{C'}$ ) is used to calculate a null ECC for each gene ( $\log(\sigma_C/\sigma_{C'})$ ); the median of these null ECCs over all the genes is used as the constant correction factor  $C = \text{median}_{\forall \text{genes}, i} \left( \log_2 \left( \frac{\sigma_{C_i}}{\sigma_{C'_i}} \right) \right)$ . (An extensive description of the correction factor is provided in the “ECC calculation details and considerations” section of the **Supplementary Information**.)

The corrected ECC for gene  $g$  is then:

$$ECC_g = \log_2 \left( \frac{\sigma_{C_g}}{\sigma_{B_g}} \right) - C$$

The computed ECC values for all yeast genes, available in **Supplementary Table 1**, were used to identify cases of presumed stabilizing selection (selection favoring a fixed non-extreme value of a trait), diversifying (disruptive) selection (selection favoring more than one extreme values of a trait; as opposed to a single fixed intermediate value), and directional (positive) selection (selection favoring a single extreme value of a trait over all other possible values of the trait). Re-computing

the ECC values for all yeast genes using the S288C reference sequences instead of the consensus sequence for the promoters of each gene yielded very similar results.

In addition to each ECC value, a Z-score and *p*-values for the confidence that the observed ECC values differ from neutrality were also calculated. For each gene's true ECC, a set of matched random ECC values were calculated, where the denominator is a set of sequences matched for Hamming distance distribution and the total number of unique sequences. The null ECC mean and standard deviation were calculated from 1,111 such simulations, and used to calculate a Z-score for how extreme the actual ECC would be under this null distribution. This Z-score acts as a signed *p*-value (negative representing divergent expression and positive representing conservation), from which *p*-values (using the 'scipy.stats.norm.sf' function on the absolute value of the Z-score in Scipy<sup>95</sup> and multiplying the function's output by 2 to get a two-sided *p*-value) (**Supplementary Table 1**).

### **Inferring expression conservation across *Saccharomyces* species using RNA-seq data and comparing with ECC values**

Published RPKM values for orthologs of *S. cerevisiae* genes in closely related *Saccharomyces* species<sup>41</sup> were obtained from the Gene Expression Omnibus (GEO) (accession GSE83120). Only genes for which expression was quantified in all species were used in subsequent analysis. RPKM values were  $\log_2$  scaled after adding a pseudo count of 2, and the variance in expression of each gene across the species was calculated. Genes were ranked by their gene expression variance, and the 2% of genes with the lowest variance were considered as having conserved gene expression levels ('expression conserved'), while the 2% with the highest variance were considered

‘expression not-conserved’. The significance of the differences was robust to the choice of these thresholds (**Supplementary Information**). To compare to ECC values, the p-value of a two-sided Wilcoxon rank-sum test was estimated comparing the ECC values for genes in the ‘expression conserved’ and ‘expression not-conserved’ categories (implemented using the *scipy.stats.ranksums* SciPy<sup>95</sup> function). To control for the dependence between expression mean and variance, the analysis was repeated using the coefficient of variation ( $P = 1.05 \times 10^{-4}$ ) and the coefficient of dispersion ( $P = 2.42 \times 10^{-4}$ ) instead of variance, yielding similar results.

### **Experimental protocol for RNA-seq measurements from 11 Ascomycota species**

RNA-seq was performed on samples from the following 11 Ascomycota yeast species: *Saccharomyces cerevisiae*, *Saccharomyces bayanus*, *Naumovozyma (Saccharomyces) castelli*, *Candida glabrata*, *Kluyveromyces lactis*, *Kluyveromyces waltii*, *Candida albicans*, *Yarrowia lipolytica*, *Schizosaccharomyces japonicus*, *Schizosaccharomyces octosporus*, and *Schizosaccharomyces pombe*. Each of the 11 species was grown in BMW medium, chosen to minimize cross-species growth differences, as previously described<sup>96</sup>. *N. castelli* was grown at 25°C while the other species were grown at 30°C. RNeasy Midi or Mini Kits (Qiagen, Valencia, CA) were used to isolate total RNA from log-phase cells by mechanical lysis using the manufacturer instructions as previously described<sup>96</sup>. dUTP strand-specific RNA-seq libraries were constructed as previously described<sup>97</sup> with the following modifications. (1) The polyA<sup>+</sup>-selected RNA was fragmented in a 40µl reaction containing 1x Fragmentation Buffer (Affymetrix) by heating at 80°C for 4 minutes followed by cleanup via ethanol precipitation for all libraries (except *Y. lipolytica*, *S. pombe*, *S. japonicus*, and *S. octosporus*; for these species, the conditions described



previously were used<sup>97</sup>), followed by cleanup via 1.8x RNAClean XP beads (Beckman Coulter Genomics). **(2)** For *C. glabrata*, *K. lactis*, *S. bayanus*, *S. pombe*, *S. japonicus*, and *S. octosporus* libraries, the adapter ligation was performed overnight at 16°C. For the rest, this was done at 16°C for 2 hours as described previously<sup>97</sup>. **(3)** Normalization was carried out based on the cDNA input and pooling of selected Illumina barcoded-adaptor-ligated cDNA products followed by gel size selection occurred as follows: range of 275 to 575 bp for pooled *C. albicans*, *K. waltii*, and *N. castellii* libraries, and 375 to 575 bp for *C. glabrata*, *K. lactis*, and *S. bayanus* libraries. For the other libraries, no pooling was performed before gel size-selection – range of 310 to 510 bp for *Y. lipolytica* and 350 to 550 bp for *S. pombe*, *S. japonicus*, and *S. octosporus*. **(4)** The final PCR product was purified by 1.8x AMPure XP beads (Beckman Coulter Genomics) followed by a second gel size-selection for the range of 300 to 575 bp for *C. albicans*, *K. waltii*, and *S. castellii* libraries, but no second gel size-selection was performed for the other libraries. The pooled final library was sequenced on one to four lanes of HiSeq2000 (Illumina) with 68 base (*Y. lipolytica* had 76 base) paired-end reads and 8 base index reads.

## **Transcript assembly, mapping and expression calculation for the 11 Ascomycota species**

### **RNA-seq**

For each of the 11 Ascomycota yeast species above, reads were assembled using Trinity<sup>98</sup>(version ‘trinityrnaseq\_r2012-05-18’) and the assembled transcripts were mapped onto the assemblies to the respective genomes using GMAP<sup>99</sup>. The Jaccard coefficient was used to join adjacent assemblies given enough connecting reads (using the Trinity default of 0.35 for the Jaccard cutoff). Finally, upon mapping all assembled transcripts, the Jaccard coefficient was used to clip assemblies which did not have enough support over a certain region. For each of the species,

assembled transcripts were mapped to the genome sequence<sup>100</sup> using BLAT<sup>89</sup>. Estimated expression values were calculated for each transcript using RSEM<sup>101</sup> (defined in RSEM as the estimate of the number of fragments that are derived from a given isoform or gene, or the expectation of the number of alignable and unfiltered fragments that are derived from an isoform or gene given the maximum likelihood abundances). Only reads mapping to the sense mRNA strand were considered. Orthology between genes in different species was used as previously described<sup>100</sup>.

### **Inferring expression conservation across Ascomycota species using our RNA-seq data and comparing with ECC values**

Estimated expression values from the 11 Ascomycota species RNA-seq data were used after removing all genes with NA values in expression for more than three species. Estimated expression values were  $\log_2$  scaled after adding a pseudo count of 1, and the variance in expression for each gene across the species was calculated. Genes were ordered by their variance in expression across the reported fungal species. Here, the 10% of genes with the lowest expression variance were considered to have ‘conserved’ expression, and the 10% with highest expression variance were considered to have expression ‘not conserved’. To compare to ECC values, the p-value of a two-sided Wilcoxon rank-sum test was estimated comparing the ECC values for genes in the ‘conserved’ and ‘not conserved’ categories (implemented using the *scipy.stats.ranksums* SciPy<sup>95</sup> function). Similar results were obtained when repeating the analysis using the coefficient of variation ( $P = 4.22 \cdot 10^{-5}$ ) and the coefficient of dispersion ( $P = 8.05 \cdot 10^{-5}$ ) instead of variance.

## **Inferring expression conservation across mammalian species using RNA-seq data and comparing with ECC values**

Ensembl Biomart<sup>102</sup> was used to find one to one orthologs of *S. cerevisiae* genes in humans (of 'Human homology type' 'ortholog\_one2one'; all 'ortholog\_one2many' and 'many2many' orthologs were excluded). For these human orthologs of yeast genes, the previously reported 'evolutionary variance' values across mammalian species from the original publication<sup>42</sup> (based on an Ornstein Uhlenbeck (OU model)<sup>42</sup>) were directly used. Here, the 25% of genes with the lowest 'evolutionary variance' were considered to have conserved expression and the top 25% were considered to be not conserved (the same thresholds used in the original study<sup>42</sup>). This was done separately for each profiled tissue (brain, heart, kidney, liver, lung and skeletal muscle). Subsequently, a human ortholog for a yeast gene was considered to have conserved (or non-conserved) expression if it was found to have conserved (or non-conserved) expression in at least one of the profiled tissues. Genes with conflicting expression conservation classes across tissues were excluded from the analysis. To compare to ECC values, the p-value of a two-sided Wilcoxon rank-sum test was estimated comparing the ECC values for genes in the "conserved" and "not conserved" categories (implemented using the *scipy.stats.ranksums* SciPy<sup>95</sup> function).

## **Quantifying sequence dissimilarity using mean Hamming distance**

For each group of orthologous yeast gene promoters (with ungapped alignments), the mean of Hamming distances between each pair of orthologous promoters across the 1,011 isolates was calculated.

## **Generation of CDC36 promoter strains by allele swapping**

Strains with a restored Upc2p binding site in the *CDC36* promoter region were obtained using a previously described CRISPR-Cas9 method<sup>103</sup>. Guide RNAs (gRNAs) were designed using the Benchling online tool (<https://www.benchling.com/>) and cloned in a pGZ110 derived plasmid<sup>104</sup>, using standard “Golden Gate Assembly”<sup>105</sup>. Plasmids carrying the gRNA and Cas9 gene were then co-transformed with a synthetic DNA fragment (ssODN) composed of a 100 bp sequence with perfect complementarity to the background promoter sequence (WE) but for the centrally-located targeted alleles that overlap the Upc2p binding site. Allele swapping was confirmed by Sanger sequencing (Macrogen, South Korea). Sequences were analyzed using the SGRP (Saccharomyces Genome Resequencing Project) BLAST server ([http://www.moseslab.csb.utoronto.ca/sgrp/blast\\_new/](http://www.moseslab.csb.utoronto.ca/sgrp/blast_new/)) and MUSCLE tool in Geneious v10.1. All primers and ssODNs used are listed in **Supplementary Table 2**.

### **RNA extraction and qPCR of *CDC36***

Gene expression analysis was performed by qPCR from cultures growth in SD medium supplemented with uracil (0.02% p/v). Samples were grown until exponential phase (OD 0.6-0.8), collected by centrifugation and treated with 10 units of Zymolyase 20T (50mg/ml) for 30 min at 37°C. RNA was extracted using E.Z.N.A Total RNA kit I (OMEGA) according to manufacturers’ instructions. Genomic DNA traces were then removed by treating samples with DNase I (Promega). RNA concentrations were estimated using a Qubit system and verified by 1.5% agarose gel. RNA extractions were performed in three biological replicates.

cDNA was synthesized using 200 units of M-MLV Reverse transcriptase (Promega), 0.5  $\mu\text{g}$  of Oligo (dT)15 primer and 1  $\mu\text{g}$  of RNA in a final volume of 25  $\mu\text{L}$  according to manufacturers' instructions. qPCR reactions were carried out using Brilliant II SYBR® Green QPCR Master Mix (Agilent Technologies) in a final volume of 10  $\mu\text{L}$ , containing 0.2  $\mu\text{M}$  of each primer and 1  $\mu\text{L}$  of the cDNA previously synthesized. qPCR reactions were carried out in three technical replicates per biological replicate using an Eco Real-Time PCR system (Illumina, Inc.) under the following conditions: 95°C for 15 min and 40 cycles at 95°C for 10 s and 58°C for 30 s. Primers used are listed in **Supplementary Table 2**. The relative expression of *CDC36* was quantified using the  $2^{-\Delta\Delta\text{Ct}}$  approach<sup>106</sup>, and normalized with two housekeeping genes as previously described<sup>107</sup>, using the median Ct of the three technical replicates for each sample. The housekeeping genes *ACT1* and *RPN2* were used as previously described<sup>108</sup>.

### **Growth curves of *CDC36* mutant and wild type alleles**

Growth curves incorporating carbon source switching from glucose to galactose were generated as previously described<sup>109</sup>. Pre-cultures were grown in YNB containing 5% glucose medium at 30°C for 24 h. Cultures were then diluted to an initial OD<sub>600nm</sub> of 0.1 in fresh YNB 5% glucose medium for an extra overnight growth. The next day, cultures were used to inoculate a 96-well plate with a final volume of 200 $\mu\text{L}$  YNB with 5% galactose with an initial OD<sub>600nm</sub> of 0.1. In parallel, a control plate containing YNB with 5% glucose was similarly inoculated. All experiments were performed in triplicate. OD<sub>600nm</sub> was monitored every 30 min using a Tecan Sunrise absorbance microplate reader (Tecan Group Ltd.). The kinetic parameters of lag phase, growth efficiency ( $\Delta\text{OD}_{600\text{ nm}}$ ) and maximum specific growth rate ( $\mu_{\text{max}}$ ) were determined as previously described<sup>110</sup>, fitting the curves with the Gompertz function using R version 3.3.2. All

growth parameters are expressed as the ratio of growth within YNB+galactose to YNB+glucose to control for phenotypic variation that results from something other than the carbon source switch.

### **Fitness responsiveness**

The empirically-determined relationships between the expression levels to organismal fitness for each of 80 genes<sup>11</sup> were re-analyzed. Published expression-to-fitness curves in glucose media for each of 80 genes were obtained from the Supplementary Data of the original publication<sup>11</sup>. For each of these curves, the total variation (**Extended Data Fig. 5**) was calculated by partitioning the expression range into 36 regular intervals (as reported in the ‘impulse fit’ of the expression-to-fitness curves in the original publication<sup>11</sup>) and summing the absolute difference in fitness at the endpoints of each partition as follows  $\sum |F_{GENE}(e_{i+1}) - F_{GENE}(e_i)|$ , for each gene’s expression-to-fitness function,  $F_{GENE}(e)$ . The same qualitative relationship between a gene’s ECC and fitness responsiveness as reported in other studies<sup>111–113</sup> was observed, including *LCB2* (ECC 2.15 and high fitness responsiveness<sup>112</sup>) and *MLS1* (ECC -1.32 and extremely low fitness responsiveness<sup>113</sup>).

### **Mutational robustness**

For every sequence, mutational robustness was defined as the fraction of sequences in its 3L mutational neighborhood that altered the expression by an amount less than  $\epsilon$ , where  $\epsilon$  is set at two times the standard deviation of expression variance across all genes with an ECC >0 (here,  $\epsilon = 0.1616$ ; ECC calculated using the 1,011 *S. cerevisiae* genomes, **Extended Data Fig. 4c**). Using different values for  $\epsilon$  yielded very similar results.

## The evolvability vector

To derive the evolvability vector for a given sequence, expression changes associated with single base changes in every possible position were sorted to obtain a monotonically increasing vector of length  $3L$  for each sequence of length  $L$  (here,  $L=80$ ;  $3L=240$ ; **Fig. 4a**, left, **Methods**). Formally, to compute an evolvability vector for a sequence  $s_0$ , for each sequence  $s_i$  in the  $3L$  mutational neighborhood of  $s_0$ , the difference between the predicted expression of  $s_i$  and that of  $s_0$  :  $d_i = f(s_i) - f(s_0)$  was calculated, where  $f(s)$  represents the predicted expression of the transformer model. The evolvability vector is defined as the vector  $D(\{d_1, d_2, \dots, d_{3L}\})$ , sorted such that  $d_i \geq d_{i-1}, \forall i$  (i.e.  $d_i$  values are in ascending order).

## Power law distribution analysis

The list of the absolute values of the evolvability vectors for all native sequences was used to define the distribution of the magnitude of the expression effect of mutations. The powerlaw<sup>114</sup> Python package was used to determine whether the data fit a power law distribution. The ‘Fit’ function with an ‘xmin’ parameter of 0.5 was used to determine the exponent and the ‘distribution\_compare’ function was used to determine the p-value for the fit (**Extended Data Fig. 4d, Supplementary Fig. 2h**).

## Characterizing the archetypal evolvability space

The evolvability vectors for a new random sample of a million sequences were used as input to an autoencoder with an archetypal regularization constraint<sup>45</sup> on the embedding layer. The autoencoder was trained using the AANet implementation made available with the publication<sup>45</sup>

with no noise added to the archetypal layer during training, a linear activation on the output layer, an equal weight of 1 on each of the loss terms (the mean squared error loss term along with the non-negativity and convexity constraints), a learning rate of 0.001, and a minibatch size of 4,096. The autoencoder accepts an evolvability vector (of length 240 for an 80bp sequence) as input to the first encoder layer, where each node in the input layer is connected to each node in the encoder layer (fully connected layer). Every layer in the autoencoder was fully connected. The encoder architecture used was [1024, 512, 256, 128, 64], where each entry corresponds to the number of nodes in the corresponding hidden layer and the decoder architecture was the encoder's mirror image. The output layer was the same shape as input layer and each node in the last decoder layer was connected to each node in the output layer. To select the optimal number of archetypes, the autoencoder was first trained for a 1,000 minibatches separately for 1 to 9 archetypes. Following the recommended approach<sup>45</sup> for picking the optimal number of archetypes, we used an elbow plot of mean squared error on the evolvability vectors (here, using native sequences) vs. the number of archetypes in the autoencoder (**Extended Data Fig. 6a**).

The autoencoder was then trained from scratch with 3 archetypes, using the full training data and parameters for 250,000 batches. Since this autoencoder aims to reconstruct the original evolvability vector for each sequence by learning feature representations after passing them through an information bottleneck, its reconstruction accuracy was first verified on the set of native yeast promoter sequences (**Extended Data Fig. 6b**, Pearson's  $r = 0.992$ ). To visualize the evolvability vectors corresponding to sequences in 2 dimensions (2D), the evolvability vectors corresponding to the three archetypes were first generated by decoding their archetypal latent space coordinates ((1,0,0), (0,1,0) and (0,0,1)) through the decoder, and MDS was performed on the



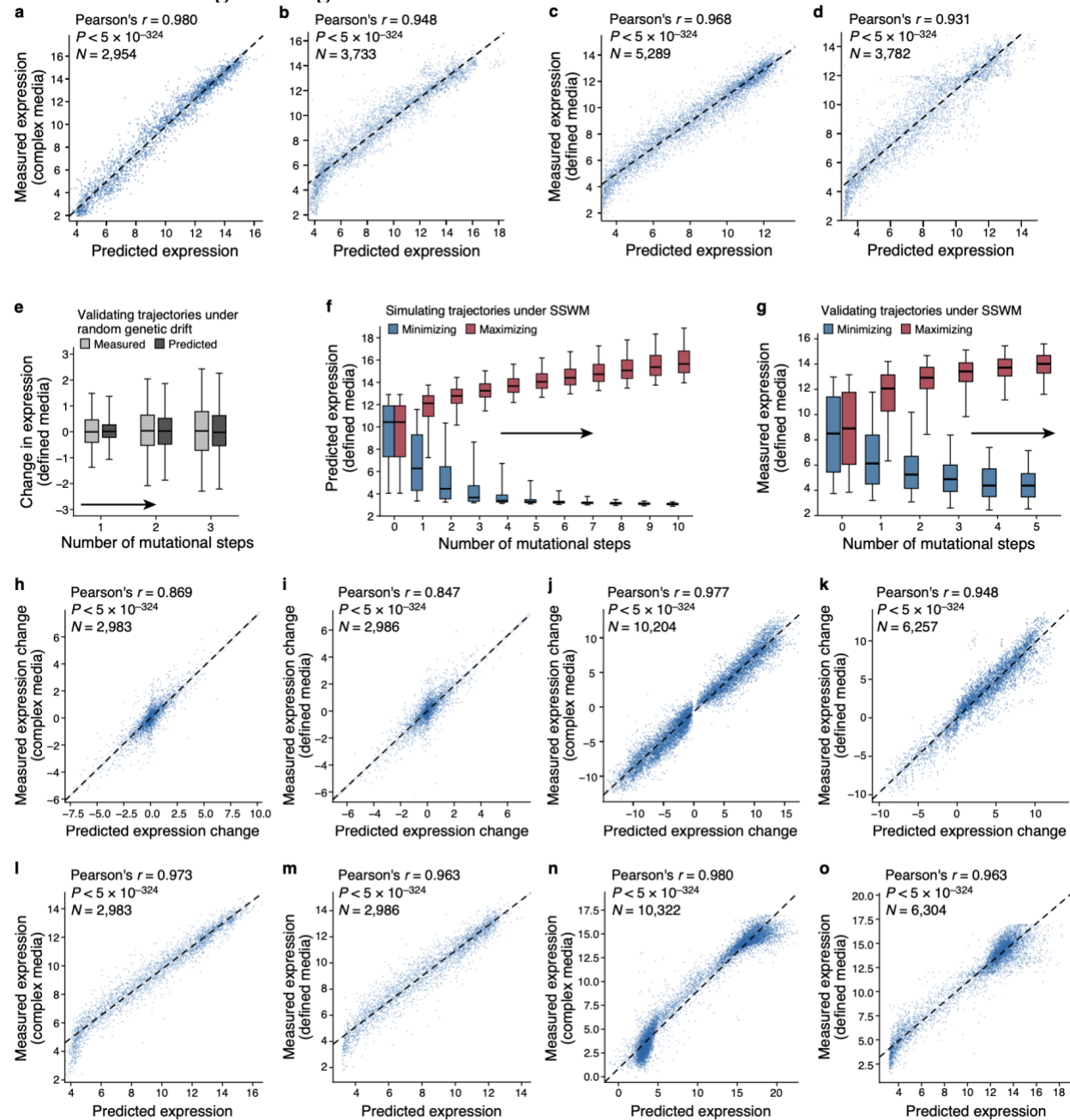
decoded evolvability vectors of the archetypes. Then, as previously described<sup>45</sup>, the encoded evolvability vector of each new sequence was projected into the 2D MDS space by representing it as a mixture of the archetypes and interpolating them between the MDS coordinates of each archetype. For every sequence, the following equivalent representations can now be computed: (i) its evolvability vector, (ii) an archetypal triplet quantifying the similarity of its encoded (latent space) evolvability vector to the three archetypes and (iii) a two-dimensional multidimensional scaling (MDS) coordinate<sup>45</sup> for visualizing the evolvability vectors. The representation of the evolvability vector for each sequence in this archetypal space is now bounded by a simplex (whose vertices correspond to the 3 evolvability archetypes). For each native and natural yeast promoter sequence from the sequence space, the archetypal triplet and MDS coordinates were inferred using its evolvability vector with this trained autoencoder. The MDS coordinates for the archetypes and the native yeast promoter sequences were used to generate the visualizations of the sequence space shown. This archetypal characterization of evolvability vectors allows the encoding and visualization of sequences by their evolvability in the context of a fitness landscape.

### **Visualizing promoter fitness landscapes**

1000 random sequences were sampled and projected onto the MDS coordinate system for visualizing the sequence space described above. The expression level of each sequence was calculated using our model, and expression values were scaled so that the minimum was 0 and maximum was 1. Previously quantified expression-to-fitness relationships<sup>11</sup> to compute fitness (fraction of wildtype growth rate) by using cubic spline interpolation (implemented using the *scipy.interpolate.CubicSpline* SciPy<sup>95</sup> function) on the expression level after scaling the measured expression-to-fitness curves to have an expression range of 0 to 1. These fitness values were then

used to generate the contour plots (implemented using the *matplotlib.pyplot.tricontourf* function; **Fig. 4e, Extended Data Fig. 7**) that visualize the fitness landscape in that gene's promoter sequence space.

## Extended Data Figures Legends



**Extended Data Fig. 1: The convolutional sequence-to-expression model generalizes reliably and helps characterize sequence trajectories under different evolutionary regimes. (a-d)** Prediction of expression from sequence in complex (YPD) (a-b) and defined (SD-Uracil) (c-d) media. Predicted ( $x$  axis) and experimentally measured ( $y$  axis) expression for (a,c) random test sequences (sampled separately from and not overlapping with the training data) and (b,d) native yeast promoter sequences containing random single base mutations. Top left: Pearson's  $r$  and associated two-tailed  $p$ -value. Compression of predictions in the lower left results from binning differences during cell sorting in different experiments (**Supplementary Notes**). **e**, Experimental validation of trajectories from simulations of random genetic drift. Distribution of measured (light grey) and predicted (dark gray) changes in expression in the defined media (SD-Uracil) ( $y$

axis) for the synthesized randomly-designed sequences (n=2,986) at each mutational step (x axis). Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **f, g**, Simulation and validation of expression trajectories under SSWM in defined media (SD-Uracil). **f**, Distribution of predicted expression levels (y axis) in defined media at each evolutionary time step (x axis) for sequences under SSWM favoring high (red) or low (blue) expression, starting with native promoter sequences (n=5,720). Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **g**, Experimentally-measured expression distribution in defined media (y axis) for the synthesized sequences (n=6,304 sequences; 637 trajectories) at each mutational step (x axis) from predicted mutational trajectories under SSWM, favoring high (red) or low (blue) expression. Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **h-o**, Experimental validation of predicted expression for sequences from the random genetic drift and SSWM simulations. Experimentally measured (y axis) and predicted (x axis) expression level (**l-o**) or expression change from the starting sequence (**h-k**) in complex (**h,j,l,n**) or defined (**i,k,m,o**) media using sequences from the random genetic drift (**Fig. 2c** and **(e)**; **h,i,l,m** here) and SSWM (**Fig. 2g** and **(g)**; **j,k,n,o** here) validation experiments. Top left: Pearson's *r* and associated two-tailed p-values.

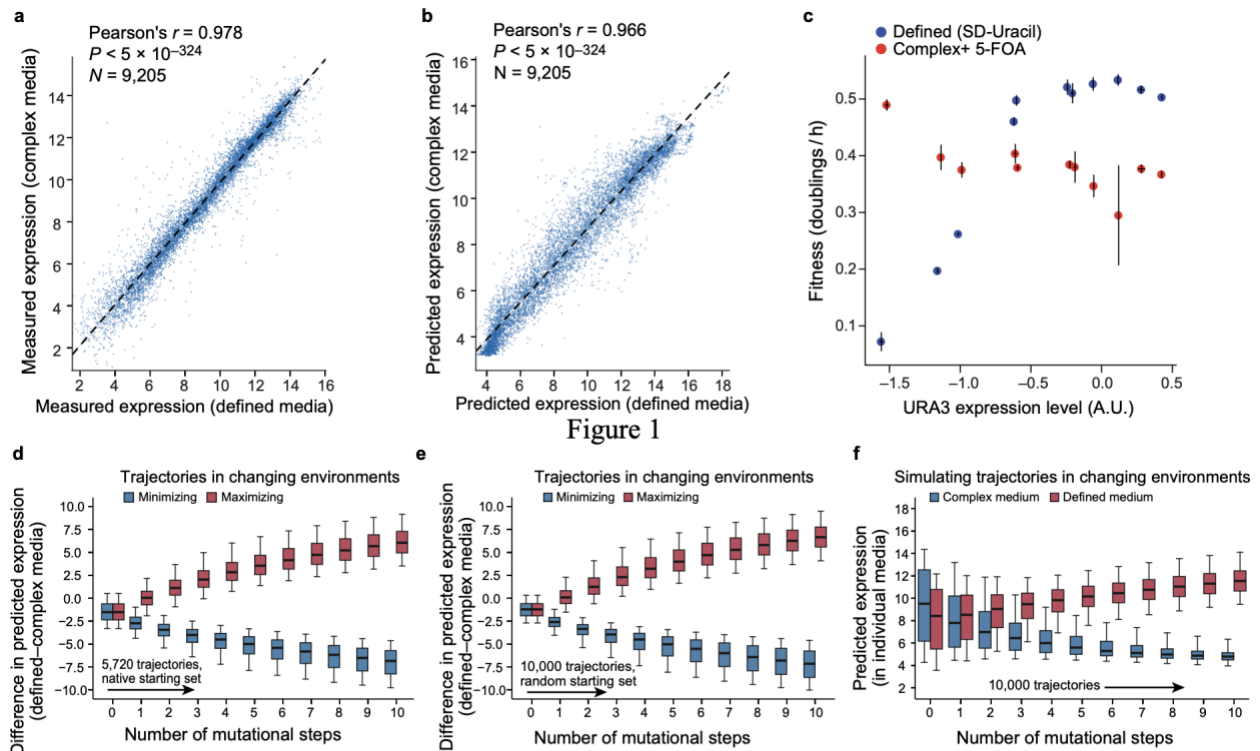


Figure 1

**g** Top 5 motifs enriched in sequences evolved to **MAXIMIZE** difference in expression between the defined and complex media

Found motif (DREME)	DREME E-value	Rank in native	Known factor	Known motif (YeTFaSCo)
	$3.1 \times 10^{-1953}$	1	Several (e.g., RPH1)	
	$1.9 \times 10^{-269}$	2	INO2/4	
	$1.1 \times 10^{-218}$	3	None	
	$9.8 \times 10^{-124}$	24	None	
	$1.7 \times 10^{-91}$	8	Several (e.g. GIS1)	

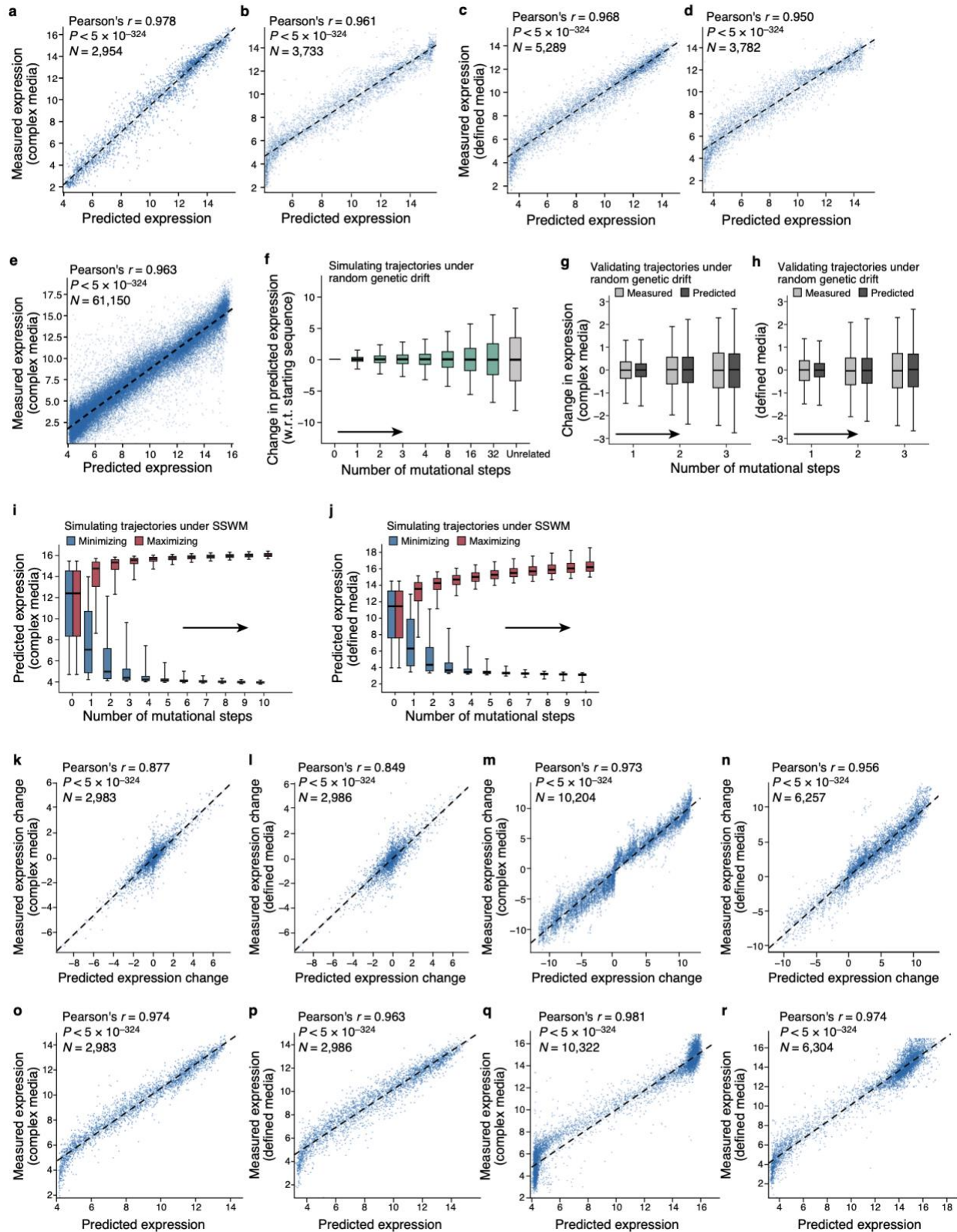
Top 5 motifs enriched in sequences evolved to **MINIMIZE** difference in expression between the defined and complex media

Found motif (DREME)	DREME E-value	Rank in native	Known factor	Known motif (YeTFaSCo)
	$1.9 \times 10^{-366}$	1	Several (e.g., PHO4)	
	$5.4 \times 10^{-250}$	5	REB1	
	$1.0 \times 10^{-184}$	24	DOT6/TOD6	
	$2.6 \times 10^{-90}$	Not found	TBF1	
	$1.6 \times 10^{-77}$	7	Several (e.g. UPC2)	

### Extended Data Fig. 2 | Characterization of sequence trajectories under strong competing selection pressures using the convolutional model.

**a,b**, Expression is highly correlated between defined and complex media. Measured (**a**) and predicted (**b**) expression in defined (*x* axis) and complex (*y* axis) media for a set of test sequences measured in both media. Top left: Pearson's *r* and associated two-tailed *p*-values. **c**, Opposing relationships between organismal fitness and *URA3* expression in two environments. Measured expression (*x* axis, using a YFP reporter) and fitness (*y* axis; when used as the promoter sequence for the *URA3* gene) for yeast with each of 11 promoters predicted to span a wide range of expression levels in complex media with 5-FOA (red), where higher expression of *URA3* is toxic due to *URA3*-mediated conversion of 5-FOA to 5-fluorouracil, and in defined media lacking uracil (blue), where *URA3* is required for uracil synthesis. Error bars: Standard error of the mean (*n*=3 replicate experiments). **d-f**, Competing expression objectives are slow to reach saturation. **d,e**, Difference in predicted expression (*y* axis) at each evolutionary time step (*x* axis) under selection to maximize (red) or

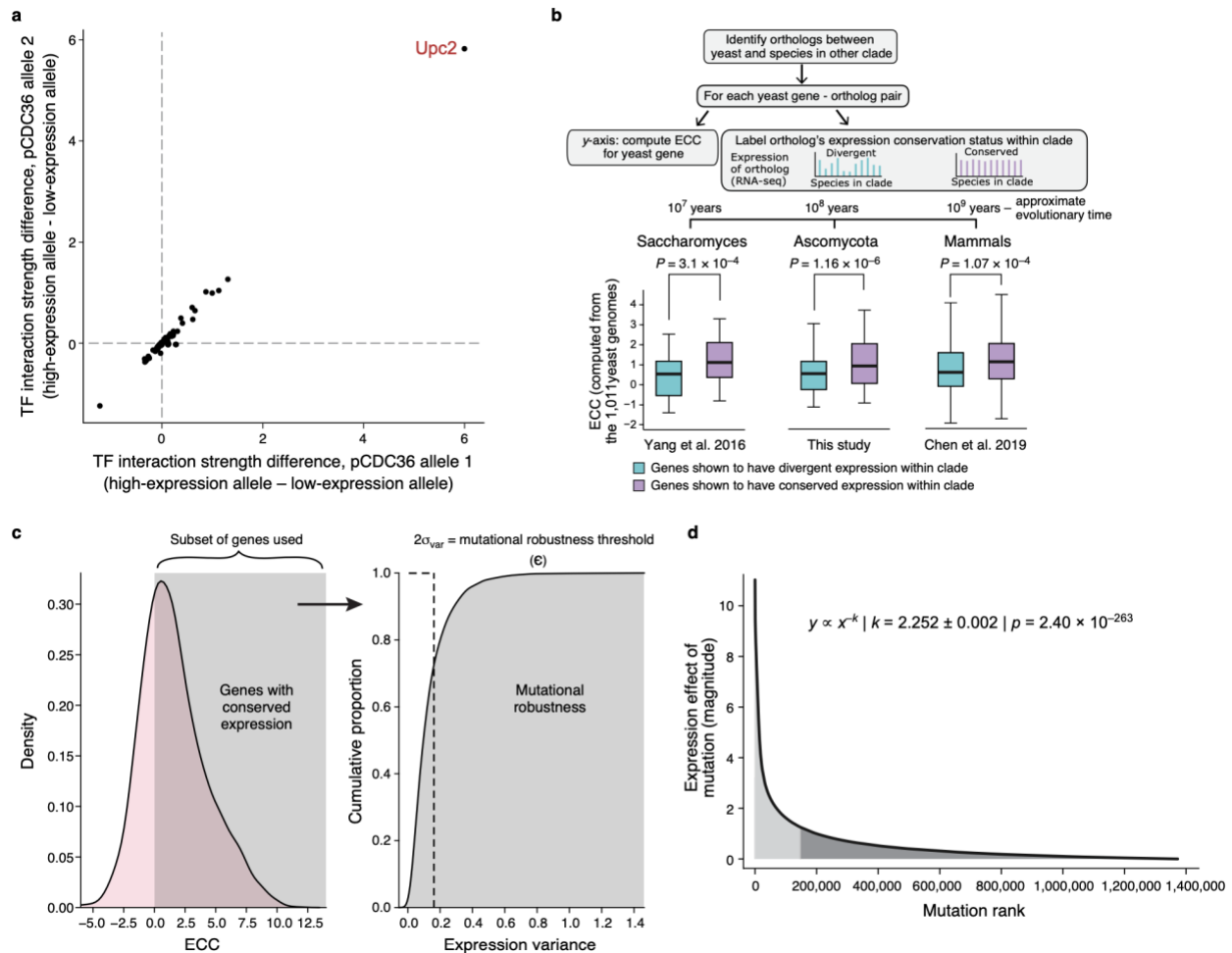
minimize (blue) the difference between expression in defined and complex media, starting with either native sequences (**d**, as **Fig. 2h**,  $n=5,720$ ) or random sequences (**e**,  $n=10,000$ ). **f**, Distribution of predicted expression ( $y$  axis) in complex (blue) and defined (red) media at each evolutionary time step ( $x$  axis) for a starting set of random sequences ( $n=10,000$ ). Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **g**, Motifs enriched within sequences evolved for competing objectives in different environments. Top five most enriched motifs, found using DREME<sup>87</sup> (**Methods**) within sequences computationally evolved from a starting set of random sequences to either maximize (left) or minimize (right) the difference in expression between defined and complex media, along with DREME E-values, the corresponding rank of the same motif when using native sequences as a starting point, the likely cognate TF and that TF's known motif.



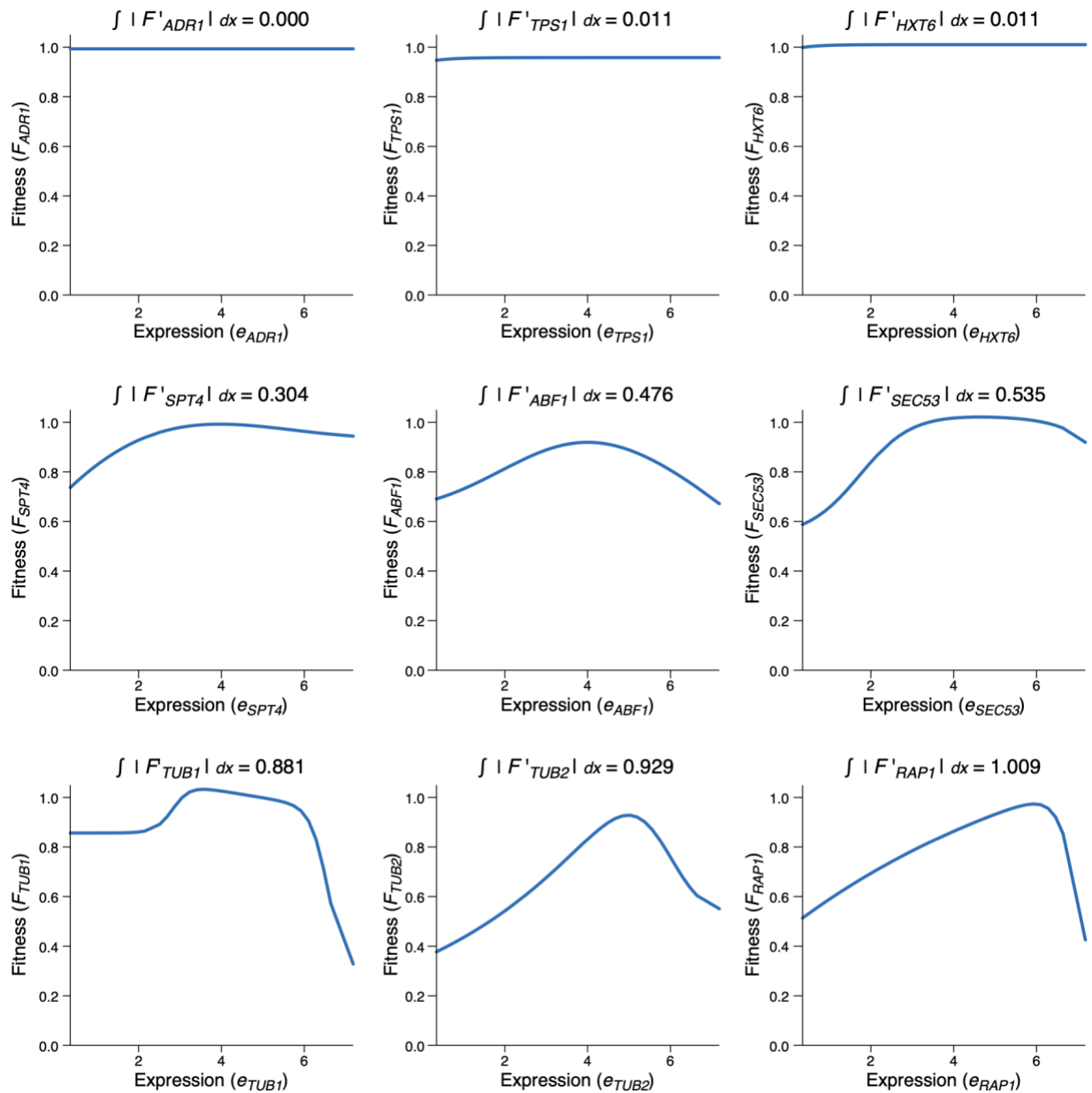
**Extended Data Fig. 3 | The transformer sequence-to-expression model generalizes reliably and helps characterize sequence trajectories under different evolutionary regimes. a-d,** Prediction of expression from sequence in the complex (a-b) and defined (c-d) media. Predicted

(*x* axis) and experimentally measured (*y* axis) expression for **(a,c)** random test sequences (sampled separately from and not overlapping with the training data) and **(b,d)** native yeast promoter sequences containing random single base mutations. Top left: Pearson's *r* and associated two-tailed *p*-value. Compression of predictions in the lower left results from binning differences during cell sorting in different experiments (**Supplementary Notes**). **e**, Predicted (*x* axis) and experimentally measured (*y* axis) expression in complex media (YPD) for all native yeast promoter sequences. Pearson's *r* and associated two-tailed *p*-values are shown. **f**, Predicted expression divergence under random genetic drift. Distribution of the change in predicted expression (*y* axis) for random starting sequences (*n*=5,720) at each mutational step (*x* axis) for trajectories simulated under random genetic drift. Silver bar: differences in expression between unrelated sequences. **g,h**, Comparison of the distribution of measured (light grey) and transformer model predicted (dark grey) changes in expression (*y* axis) in complex media (**g**, *n*=2,983) and defined media (**h**, *n*=2,986) for synthesized randomly-designed sequences at each mutational step (*x* axis). **i,j** Predicted expression evolution under SSWM. Distribution of predicted expression levels (*y* axis) in complex media (**i**, *n*=10,322) and defined media (**j**, *n*=6,304) at each mutational step (*x* axis) for sequence trajectories under SSWM favoring high (red) or low (blue) expression, starting with 5,720 native promoter sequences. (**f-j**) Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **k-r**, Comparison of model predicted expression for sequences synthesized previously for the random genetic drift and SSWM analyses. Experimentally measured (*y* axis) and transformer model predicted (*x* axis) expression level (**o-r**) or expression change from the starting sequence (**k-n**) in complex (**k,m,o,q**) or defined (**l,n,p,r**) media using sequences from the random genetic drift (**Fig. 2c** and **(Extended Data Fig. 1e)**; **k,l,o,p** here) and SSWM (**Fig. 2g** and **(Extended Data Fig. 1g)**; **m,n,q,r** here) validation experiments. Top left: Pearson's *r* and associated two-tailed *p*-values.

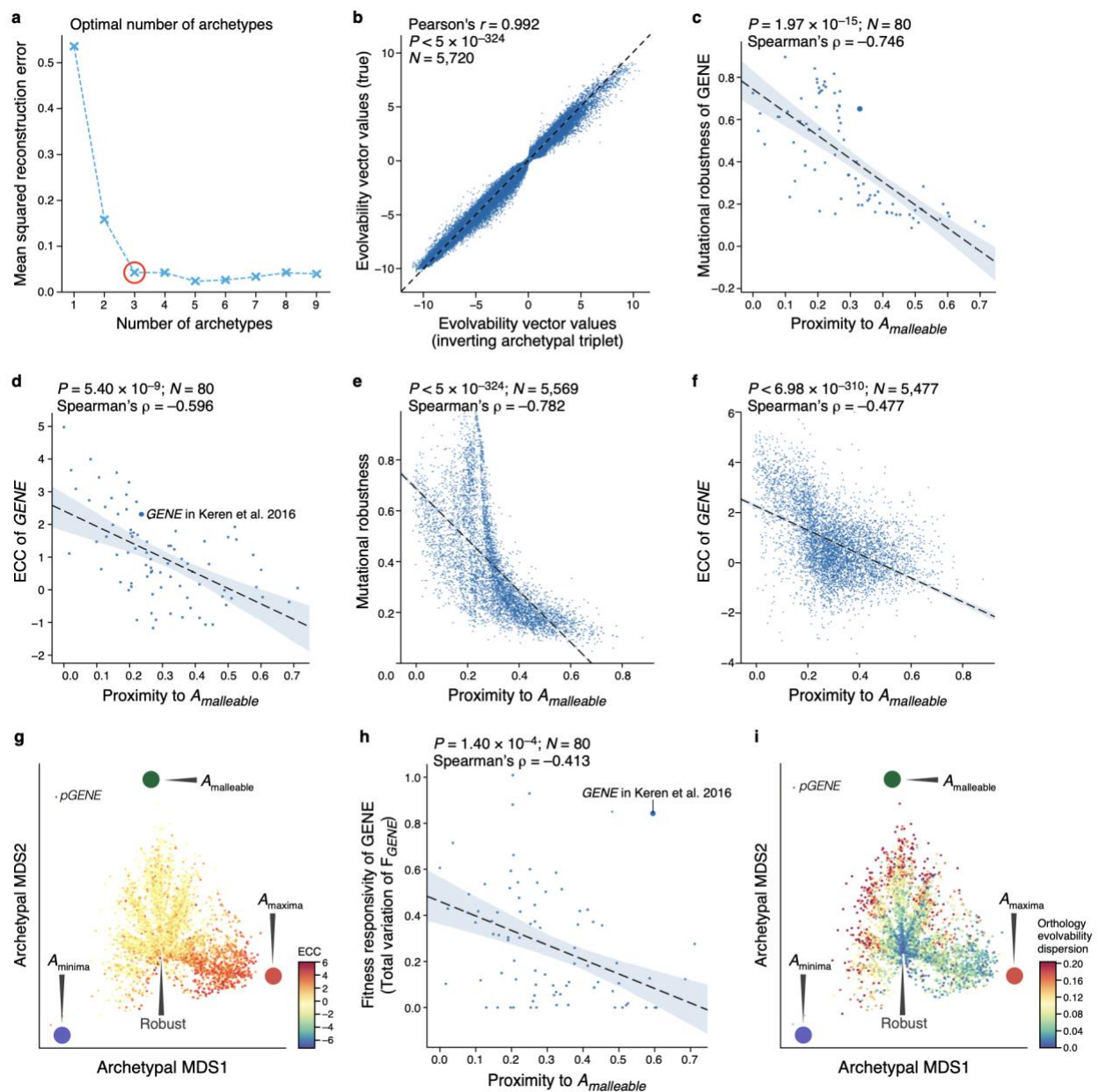




**Extended Data Fig. 4 | Signatures of stabilizing selection on gene expression detected from regulatory DNA across natural populations.** **a**, Expression-altering alleles in the CDC36 promoter are attributed primarily to altered UPC2 binding. TF interaction strength<sup>26</sup> (expression attributable to each TF) difference between the high and low alleles (each point is a TF) for each of two low expression alleles (allele 1:  $x$  axis; allele 2:  $y$  axis). Each low-expressing allele is compared to the high-expression allele with the most similar sequence (across all promoter sequences analyzed from the 1,011 strains;  $e_{TF,A_{high}} - e_{TF,A_{low}}$ ). **b**, Distribution of ECC ( $y$  axis, calculated from 1,011 *S. cerevisiae* genomes, top left) for *S. cerevisiae* genes whose orthologs have divergent (blue) or conserved (purple) expression (within *Saccharomyces* (left,  $n=4,191$ ), Ascomycota (middle,  $n=4,910$ ), or mammals (right,  $n=199$ ) (as determined by cross species RNA-seq, top right).  $p$ -values: two-sided Wilcoxon rank-sum test. Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **c**, Determination of expression change threshold for defining a "tolerated mutation" to compute mutational robustness. We used all genes with an ECC consistent with stabilizing selection ( $ECC > 0$ ; left), calculated the variance in predicted expression across the 1011 yeast strains for each gene, and chose the tolerable mutation threshold,  $\epsilon$ , as two standard deviations of the distribution of the variance (right). ~73% of genes with  $ECC > 0$  had an expression variation lower than  $\epsilon$ . **d**, Distribution of the effects of mutations (magnitude) on expression for all native regulatory sequences follows a power law with an exponent of 2.252. Shaded regions are equal in area.

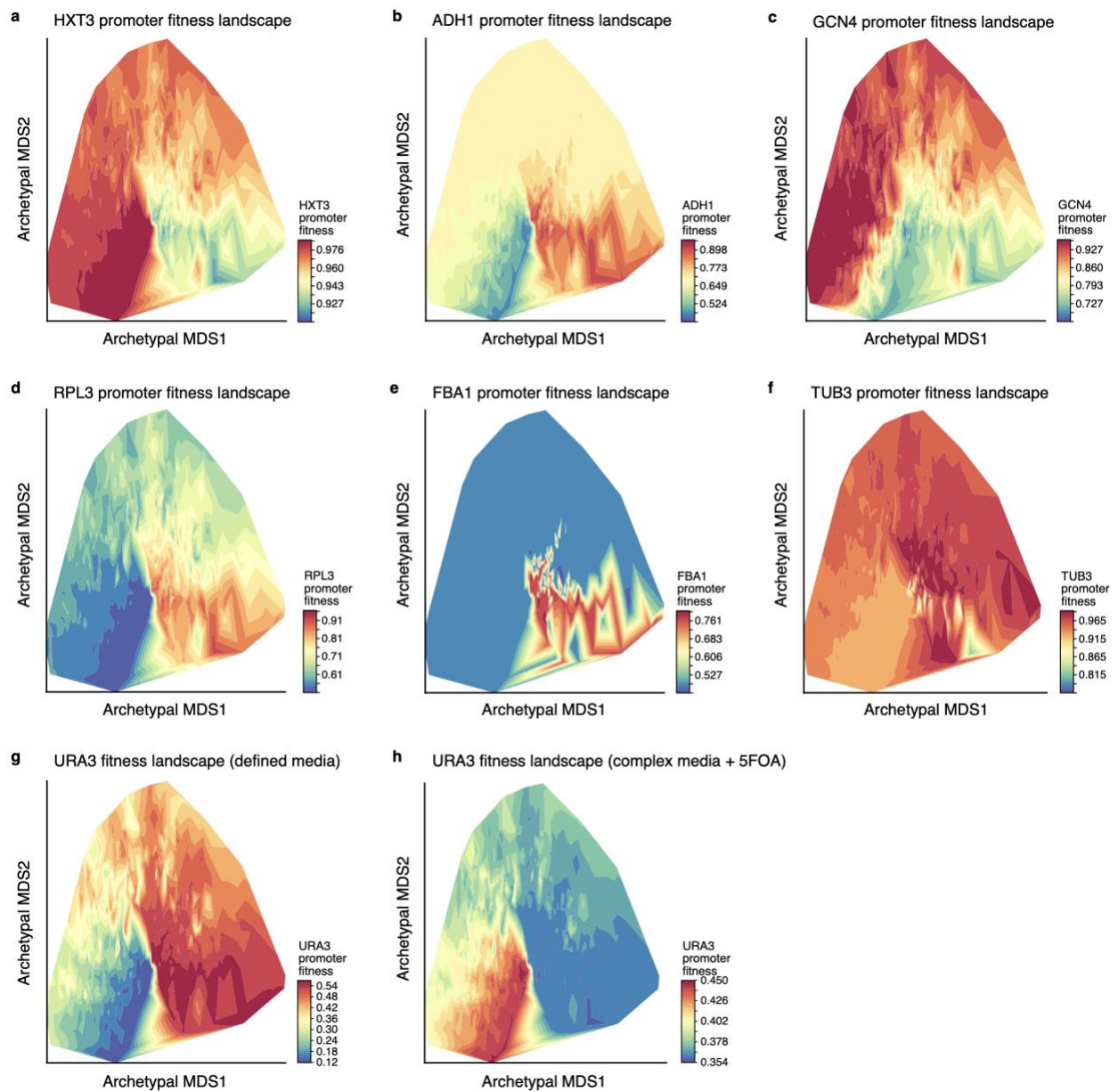


**Extended Data Fig. 5 | Fitness responsivity of a gene as the total variation of its expression-to-fitness relationship  $F_{GENE}$  curves.** Expression ( $x$  axis) and fitness ( $y$  axis) levels for different promoter variants for each select gene fit from experimental measurements by Keren et al<sup>11</sup>. Fitness responsivity calculated as the total variation in each curve is noted above each panel.



**Extended Data Fig. 6 | Analysis of regulatory evolvability reveals sequence-encoded signatures of expression conservation from solitary sequences.** **a**, Selection of optimal number of archetypes. Mean-square-reconstruction error ( $y$  axis) for reconstructing the evolvability vectors from the embeddings learned by the autoencoder for an increasing number of archetypes ( $x$  axis). Red circle: optimal number of archetypes selected as prescribed<sup>45</sup> by the “elbow method”. **b**, The archetypal embeddings learned by the autoencoder accurately capture evolvability vectors. Original ( $y$  axis) and reconstructed ( $x$  axis) expression changes (the values in the evolvability vectors) for each native sequence (none seen by the autoencoder in training). Top left: Pearson’s  $r$  and associated two-tailed  $p$ -values. **c-f**, Evolvability space captures regulatory sequences’ evolutionary properties. Proximity to the malleable archetype ( $A_{malleable}$ ) ( $x$  axis) and mutational robustness (**c,e**  $y$  axis) or ECC (**d,f**  $y$  axis) for all yeast genes (**e,f**) or the gene for which fitness responsiveness was quantified (**c,d**). Top right: Spearman’s  $\rho$  and associated two-sided  $p$ -value. “L”-shape of relationship in **e** results from the robust cleft,  $A_{maxima}$ , and

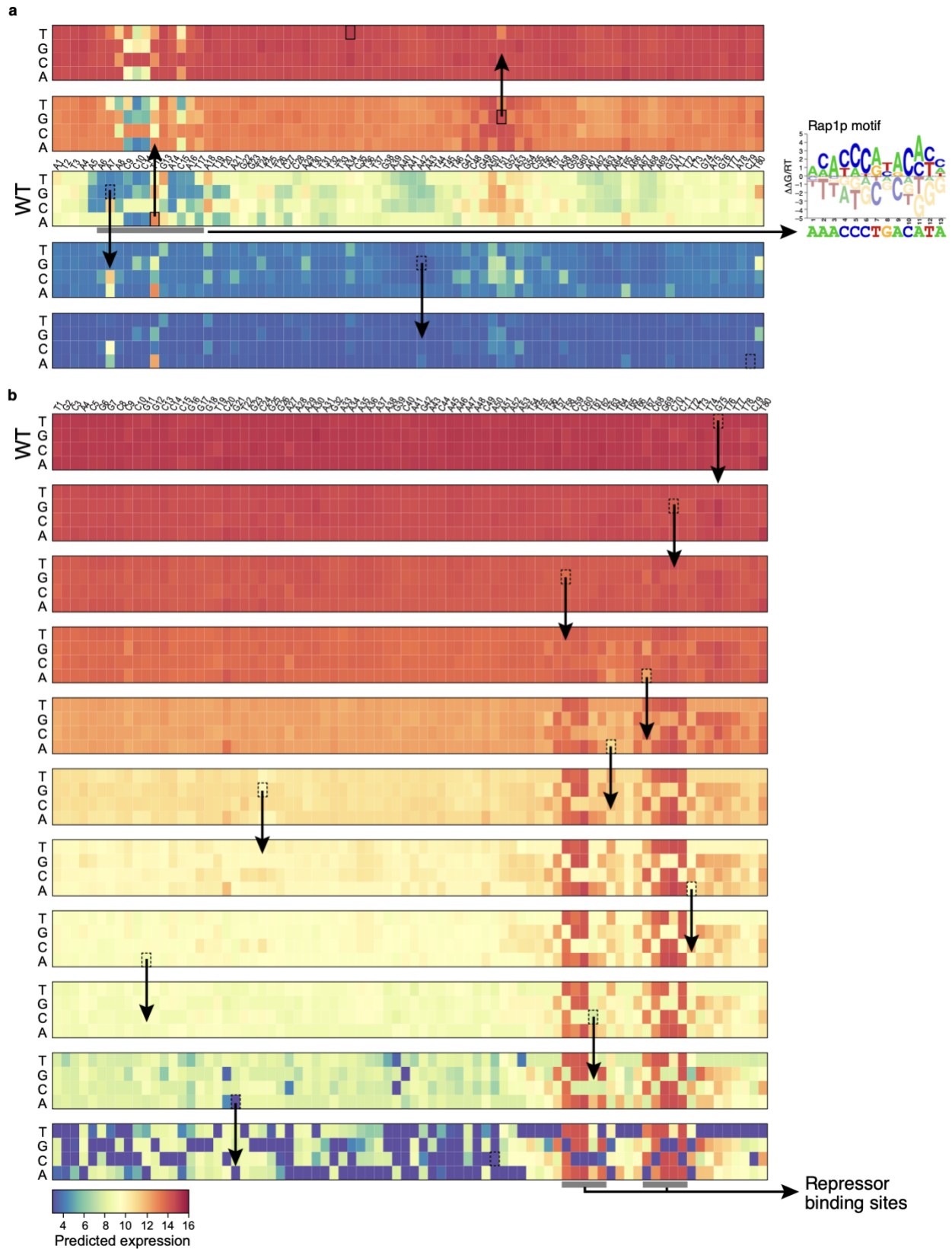
$A_{\text{minima}}$  all being distal to  $A_{\text{malleable}}$  (left side of plot). **g**, All native (S288C reference) promoter sequences (points) projected onto the archetypal evolvability space learned from random sequences; colored by their ECC. Large colored circles: evolvability archetypes. **h**, The proximity to the malleable archetype ( $x$  axis) and fitness responsiveness ( $y$  axis) for the 80 genes with measured fitness responsiveness. Top right: Spearman's  $\rho$  and associated two-tailed  $p$ -values. Light blue error band: 95% confidence interval. **i**, All native (S288C reference) promoter sequences (points) projected on the evolvability space learned from random sequences; colored by their mean pairwise distance in the archetypal evolvability space between all promoter alleles across the 1,011 yeast isolates for that gene (ortholog evolvability dispersion). Large colored circles: evolvability archetypes.



**Extended Data Fig. 7 | Visualizing promoter fitness landscapes in sequence space.**

Visualizing the fitness landscapes for the promoters of *HXT3* (a), *ADH1* (b), *GCN4* (c), *RPL3* (d), *FBA1* (e), *TUB3* (f), *URA3* (in defined media) (g), *URA3* (in complex media + 5FOA) (h).

1000 promoter sequences represented by their evolvability vectors projected onto the 2D archetypal evolvability space and colored by their associated fitness as reflected by their predicted growth rate relative to wildtype (color, **Methods**), estimated by first mapping sequences to expression with our model and then expression to fitness as measured and estimated previously<sup>11</sup>.



**Extended Data Fig. 8 | In silico mutagenesis (ISM) of malleable and robust promoters.** SSWM trajectories for (a) *DBP7*, a malleable promoter, and (b) *UTH1*, a robust promoter. Each

subplot shows the *in silico* mutagenesis effects for how expression level (color) changes when mutating each position (*x* axis) to each of the four bases (*y* axis) of each sequence (subplots) in the trajectories. The DNA sequence is indicated above each wildtype subplot (indicated with “WT” at left). Arrows indicate the mutations selected at each step, which always correspond to the mutation of maximal effect; increasing expression goes up the figure from wildtype and decreasing expression goes down. Part of the malleability of the *DBP7* promoter results from an intermediate-affinity Rap1p binding site (gray bar). The first mutations in increasing- and decreasing-expression trajectories either increase or decrease (respectively) the affinity of this site. The *UTH1* promoter changes gradually in expression and evolves proximal repressor binding sites to dampen expression (gray bars).

## References

1. Wittkopp, P. J. & Kalay, G. Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence. *Nat. Rev. Genet.* **13**, 59–69 (2011).
2. Hill, M. S., Vande Zande, P. & Wittkopp, P. J. Molecular and evolutionary processes generating variation in gene expression. *Nature Reviews Genetics* 1–13 (2020).
3. Fuqua, T. *et al.* Dense and pleiotropic regulatory information in a developmental enhancer. *Nature* 1–5 (2020).
4. de Visser, J. A. G. M. & Krug, J. Empirical fitness landscapes and the predictability of evolution. *Nat. Rev. Genet.* **15**, 480–490 (2014).
5. Kondrashov, D. A. & Kondrashov, F. A. Topological features of rugged fitness landscapes in sequence space. *Trends in Genetics* **31**, 24–33 (2015).
6. de Visser, J. A. G. M., Elena, S. F., Fragata, I. & Matuszewski, S. The utility of fitness landscapes and big data for predicting evolution. *Heredity (Edinb)* **121**, 401–405 (2018).
7. Weirauch, M. T. & Hughes, T. R. Conserved expression without conserved regulatory sequence: the more things change, the more they stay the same. *Trends Genet.* **26**, 66–74 (2010).
8. Orr, H. A. The genetic theory of adaptation: a brief history. *Nature Reviews Genetics* **6**, 119–127 (2005).
9. Weinreich, D. M., Lan, Y., Wylie, C. S. & Heckendorn, R. B. Should evolutionary geneticists worry about higher-order epistasis? *Current Opinion in Genetics & Development* **23**, 700–707 (2013).
10. Venkataram, S. *et al.* Development of a Comprehensive Genotype-to-Fitness Map of Adaptation-Driving Mutations in Yeast. *Cell* **166**, 1585-1596.e22 (2016).
11. Keren, L. *et al.* Massively Parallel Interrogation of the Effects of Gene Expression Levels on Fitness. *Cell* **166**, 1282-1294.e18 (2016).
12. Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature* **533**, 397–401 (2016).
13. Ogden, P. J., Kelsic, E. D., Sinai, S. & Church, G. M. Comprehensive AAV capsid fitness landscape reveals a viral gene and enables machine-guided design. *Science* **366**, 1139–1143 (2019).
14. Pitt, J. N. & Ferré-D’Amaré, A. R. Rapid construction of empirical RNA fitness landscapes. *Science* **330**, 376–379 (2010).
15. Shultzaberger, R. K., Malashock, D. S., Kirsch, J. F. & Eisen, M. B. The Fitness Landscapes of cis-Acting Binding Sites in Different Promoter and Environmental Contexts. *PLOS Genetics* **6**, e1001042 (2010).
16. Mustonen, V., Kinney, J., Callan, C. G. & Lässig, M. Energy-dependent fitness: a quantitative model for the evolution of yeast transcription factor binding sites. *Proc. Natl. Acad. Sci. U.S.A.* **105**, 12376–12381 (2008).
17. Hartl, D. L. What Can We Learn From Fitness Landscapes? *Curr Opin Microbiol* **0**, 51–57 (2014).



18. Otwinowski, J. & Nemenman, I. Genotype to phenotype mapping and the fitness landscape of the *E. coli* lac promoter. *PLoS ONE* **8**, e61570 (2013).
19. Sinai, S. & Kelsic, E. D. A primer on model-guided exploration of fitness landscapes for biological sequence design. *arXiv:2010.10614 [cs, q-bio]* (2020).
20. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods* **12**, 931–934 (2015).
21. Avsec, Ž. *et al.* Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat. Genet.* **53**, 354–366 (2021).
22. Shrikumar, A., Greenside, P. & Kundaje, A. Learning Important Features Through Propagating Activation Differences. *arXiv [cs.CV]* (2017).
23. Avsec, Ž. *et al.* Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* **18**, 1196–1203 (2021).
24. Fragata, I., Blanckaert, A., Louro, M. A. D., Liberles, D. A. & Bank, C. Evolution in the light of fitness landscape theory. *Trends in Ecology & Evolution* **34**, 69–82 (2019).
25. Payne, J. L. & Wagner, A. The causes of evolvability and their evolution. *Nature Reviews Genetics* **20**, 24–38 (2019).
26. de Boer, C. G. *et al.* Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nat. Biotechnol.* **38**, 56–65 (2020).
27. Crocker, J. *et al.* Low affinity binding site clusters confer hox specificity and regulatory robustness. *Cell* **160**, 191–203 (2015).
28. Habib, N., Wapinski, I., Margalit, H., Regev, A. & Friedman, N. A functional selection model explains evolutionary robustness despite plasticity in regulatory networks. *Mol. Syst. Biol.* **8**, 619 (2012).
29. Gillespie, J. H. Molecular Evolution Over the Mutational Landscape. *Evolution* **38**, 1116–1129 (1984).
30. Jerison, E. R. & Desai, M. M. Genomic investigations of evolutionary dynamics and epistasis in microbial evolution experiments. *Curr. Opin. Genet. Dev.* **35**, 33–39 (2015).
31. Sæther, B.-E. & Engen, S. The concept of fitness in fluctuating environments. *Trends Ecol. Evol.* **30**, 273–281 (2015).
32. Vaswani, A. *et al.* Attention is All you Need. in *Advances in Neural Information Processing Systems 30* (eds. Guyon, I. *et al.*) 5998–6008 (Curran Associates, Inc., 2017).
33. Weirauch, M. T. *et al.* Evaluation of methods for modeling transcription factor sequence specificity. *Nat. Biotechnol.* **31**, 126–134 (2013).
34. Yang, N. & Bielawski, N. Statistical methods for detecting molecular adaptation. *Trends Ecol. Evol. (Amst.)* **15**, 496–503 (2000).
35. Moses, A. M. Statistical tests for natural selection on regulatory regions based on the strength of transcription factor binding sites. *BMC Evolutionary Biology* **9**, 286 (2009).
36. Rifkin, S. A., Houle, D., Kim, J. & White, K. P. A mutation accumulation assay reveals a broad capacity for rapid evolution of gene expression. *Nature* **438**, 220–223 (2005).

37. Peter, J. *et al.* Genome evolution across 1,011 *Saccharomyces cerevisiae* isolates. *Nature* **556**, 339–344 (2018).
38. Erb, I. & van Nimwegen, E. Transcription factor binding site positioning in yeast: proximal promoter motifs characterize TATA-less promoters. *PLoS One* **6**, e24279 (2011).
39. Gilad, Y., Oshlack, A. & Rifkin, S. A. Natural selection on gene expression. *Trends Genet.* **22**, 456–461 (2006).
40. Alhusaini, N. & Collier, J. The deadenylase components Not2p, Not3p, and Not5p promote mRNA decapping. *RNA* **22**, 709–721 (2016).
41. Yang, J.-R., Maclean, C. J., Park, C., Zhao, H. & Zhang, J. Intra and Interspecific Variations of Gene Expression Levels in Yeast Are Largely Neutral: (Nei Lecture, SMCBE 2016, Gold Coast). *Mol. Biol. Evol.* **34**, 2125–2139 (2017).
42. Chen, J. *et al.* A quantitative framework for characterizing the evolutionary history of mammalian gene expression. *Genome Res.* **29**, 53–63 (2019).
43. Payne, J. L. & Wagner, A. Mechanisms of mutational robustness in transcriptional regulation. *Front. Genet.* **6**, (2015).
44. Shoval, O. *et al.* Evolutionary trade-offs, Pareto optimality, and the geometry of phenotype space. *Science* **336**, 1157–1160 (2012).
45. van Dijk, D. *et al.* Finding Archetypal Spaces Using Neural Networks. in (IEEE, 2019).
46. He, X., Duque, T. S. P. C. & Sinha, S. Evolutionary origins of transcription factor binding site clusters. *Mol Biol Evol* **29**, 1059–1070 (2012).
47. Cliften, P. F. *et al.* Surveying *Saccharomyces* genomes to identify functional elements by comparative DNA sequence analysis. *Genome Res.* **11**, 1175–1186 (2001).
48. Heinz, S., Romanoski, C. E., Benner, C. & Glass, C. K. The selection and function of cell type-specific enhancers. *Nature Reviews Molecular Cell Biology* **16**, 144–154 (2015).
49. Lehner, B. Selection to minimise noise in living systems and its implications for the evolution of gene expression. *Molecular Systems Biology* **4**, 170 (2008).
50. Metzger, B. P. H., Yuan, D. C., Gruber, J. D., Duveau, F. & Wittkopp, P. J. Selection on noise constrains variation in a eukaryotic promoter. *Nature* **521**, 344–347 (2015).
51. Kosuri, S. *et al.* Composability of regulatory sequences controlling transcription and translation in *Escherichia coli*. *Proc. Natl. Acad. Sci. U. S. A.* **110**, 14024–14029 (2013).
52. Shalem, O. *et al.* Systematic dissection of the sequence determinants of gene 3' end mediated expression control. *PLoS Genet.* **11**, e1005147 (2015).
53. Kinney, J. B., Murugan, A., Callan, C. G., Jr & Cox, E. C. Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 9158–9163 (2010).
54. Sharon, E. *et al.* Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters. *Nat. Biotechnol.* **30**, 521–530 (2012).
55. Melnikov, A. *et al.* Systematic dissection and optimization of inducible enhancers in human cells using a massively parallel reporter assay. *Nature biotechnology* **30**, 271–277 (2012).

56. Kwasnieski, J. C., Mogno, I., Myers, C. A., Corbo, J. C. & Cohen, B. A. Complex effects of nucleotide variants in a mammalian cis-regulatory element. *Proceedings of the National Academy of Sciences of the United States of America* **109**, 19498–19503 (2012).
57. Kircher, M. *et al.* Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. *Nature Communications* **10**, 3583 (2019).
58. Townsley, K. G., Brennand, K. J. & Huckins, L. M. Massively parallel techniques for cataloguing the regulome of the human brain. *Nat. Neurosci.* **23**, 1509–1521 (2020).
59. Renganaath, K. *et al.* Systematic identification of cis-regulatory variants that cause gene expression differences in a yeast cross. *Elife* **9**, (2020).
60. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat Methods* **9**, 357–359 (2012).
61. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* **33**, 831–838 (2015).
62. T, C. *et al.* Opportunities and obstacles for deep learning in biology and medicine. *J R Soc Interface* **15**, (2018).
63. Avsec, Ž. *et al.* The Kipoi repository accelerates community exchange and reuse of predictive models for genomics. *Nature Biotechnology* **37**, 592–600 (2019).
64. Quang, D. & Xie, X. FactorNet: A deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods (San Diego, Calif.)* **166**, 40–47 (2019).
65. Shrikumar, A., Greenside, P. & Kundaje, A. Reverse-complement parameter sharing improves deep learning models for genomics. *bioRxiv* 103663 (2017).
66. Morrow, A. *et al.* Convolutional Kitchen Sinks for Transcription Factor Binding Site Prediction. *arXiv:1706.00125 [q-bio]* (2017).
67. Kelley, D. R., Snoek, J. & Rinn, J. L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* **26**, 990–999 (2016).
68. Koo, P. K., Majdandzic, A., Ploenzke, M., Anand, P. & Paul, S. B. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLoS Comput. Biol.* **17**, e1008925 (2021).
69. Quang, D. & Xie, X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* **44**, e107 (2016).
70. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv [cs.LG]* (2014).
71. Abadi, M. *et al.* TensorFlow: A system for large-scale machine learning. *arXiv:1605.08695 [cs]* (2016).
72. Jouppi, N. P. *et al.* In-Datacenter Performance Analysis of a Tensor Processing Unit. *arXiv:1704.04760 [cs]* (2017).
73. Li, J., Pu, Y., Tang, J., Zou, Q. & Guo, F. DeepATT: a hybrid category attention neural network for identifying functional effects of DNA sequences. *Brief. Bioinform.* (2020) doi:10.1093/bib/bbaa159.

74. Ullah, F. & Ben-Hur, A. A self-attention model for inferring cooperativity between regulatory features. *Nucleic Acids Res.* (2021) doi:10.1093/nar/gkab349.
75. Clauwaert, J., Menschaert, G. & Waegeman, W. Explainability in transformer models for functional genomics. *Brief. Bioinform.* (2021) doi:10.1093/bib/bbab060.
76. Hinton, G. & Tieleman, T. Lecture 6.5---RmsProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning* (2012).
77. Sinai, S. *et al.* AdaLead: A simple and robust adaptive greedy search algorithm for sequence design. *arXiv:2010.02141 [cs, math, q-bio]* (2020).
78. Linder, J., Bogard, N., Rosenberg, A. B. & Seelig, G. A Generative Neural Network for Maximizing Fitness and Diversity of Synthetic DNA and Protein Sequences. *Cell Systems* **11**, 49-62.e16 (2020).
79. Brookes, David and Park, Hahnbeom and Listgarten, Jennifer. Conditioning by adaptive sampling for robust design. *Proceedings of Machine Learning Research* (2020).
80. Killoran, N., Lee, L. J., DeLong, A., Duvenaud, D. & Frey, B. J. Generating and designing DNA with deep generative models. *arXiv:1712.06148 [cs, q-bio, stat]* (2017).
81. Fortin, F.-A., Rainville, F.-M. D., Gardner, M.-A., Parizeau, M. & Gagné, C. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research* **13**, 2171–2175 (2012).
82. Jaeger, S. A. *et al.* Conservation and regulatory associations of a wide affinity range of mouse transcription factor binding sites. *Genomics* **95**, 185–195 (2010).
83. Tanay, A. Extensive low-affinity transcriptional interactions in the yeast genome. *Genome Res.* **16**, 962–972 (2006).
84. Sniegowski, P. D. & Gerrish, P. J. Beneficial mutations and the dynamics of adaptation in asexual populations. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **365**, 1255–1263 (2010).
85. Szendro, I. G., Franke, J., Visser, J. A. G. M. de & Krug, J. Predictability of evolution depends nonmonotonically on population size. *PNAS* **110**, 571–576 (2013).
86. Orr, H. A. The Population Genetics of Adaptation: The Adaptation of Dna Sequences. *Evolution* **56**, 1317–1330 (2002).
87. Bailey, T. L. DREME: motif discovery in transcription factor ChIP-seq data. *Bioinformatics (Oxford, England)* **27**, 1653–1659 (2011).
88. de Boer, C. G. & Hughes, T. R. YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities. *Nucleic acids research* **40**, D169-79 (2012).
89. Kent, W. J. BLAT—The BLAST-Like Alignment Tool. *Genome Res.* **12**, 656–664 (2002).
90. Cherry, J. M. *et al.* Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic Acids Res* **40**, D700–D705 (2012).
91. Smith, J. D., McManus, K. F. & Fraser, H. B. A novel test for selection on cis-regulatory elements reveals positive and negative selection acting on mammalian transcriptional enhancers. *Mol. Biol. Evol.* **30**, 2509–2518 (2013).
92. Liu, J. & Robinson-Rechavi, M. Robust inference of positive selection on regulatory sequences in the human brain. *Sci Adv* **6**, (2020).

93. Rice, D. P. & Townsend, J. P. A test for selection employing quantitative trait locus and mutation accumulation data. *Genetics* **190**, 1533–1545 (2012).
94. Denver, D. R., Morris, K., Lynch, M. & Thomas, W. K. High mutation rate and predominance of insertions in the *Caenorhabditis elegans* nuclear genome. *Nature* **430**, 679–682 (2004).
95. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* **17**, 261–272 (2020).
96. Thompson, D. A. *et al.* Evolutionary principles of modular gene regulation in yeasts. *eLife* **2**, e00603 (2013).
97. Yassour, M. *et al.* Strand-specific RNA sequencing reveals extensive regulated long antisense transcripts that are conserved across yeast species. *Genome Biology* **11**, R87 (2010).
98. Grabherr, M. G. *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology* **29**, 644–652 (2011).
99. Wu, T. D. & Watanabe, C. K. GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics* **21**, 1859–1875 (2005).
100. Wapinski, I., Pfeffer, A., Friedman, N. & Regev, A. Natural history and evolutionary principles of gene duplication in fungi. *Nature* **449**, 54–61 (2007).
101. Li, B. & Dewey, C. N. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* **12**, 323 (2011).
102. Yates, A. D. *et al.* Ensembl 2020. *Nucleic Acids Res* **48**, D682–D688 (2020).
103. DiCarlo, J. E. *et al.* Genome engineering in *Saccharomyces cerevisiae* using CRISPR-Cas systems. *Nucleic Acids Res.* **41**, 4336–4343 (2013).
104. Fleiss, A. *et al.* Reshuffling yeast chromosomes with CRISPR/Cas9. *PLoS Genet.* **15**, e1008332 (2019).
105. Horwitz, A. A. *et al.* Efficient Multiplexed Integration of Synergistic Alleles and Metabolic Pathways in Yeasts via CRISPR-Cas. *Cell Syst* **1**, 88–96 (2015).
106. Livak, K. J. & Schmittgen, T. D. Analysis of relative gene expression data using real-time quantitative PCR and the 2(-Delta Delta C(T)) Method. *Methods* **25**, 402–408 (2001).
107. Vandesompele, J. *et al.* Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes. *Genome Biol.* **3**, RESEARCH0034 (2002).
108. Teste, M.-A., Duquenne, M., François, J. M. & Parrou, J.-L. Validation of reference genes for quantitative expression analysis by real-time RT-PCR in *Saccharomyces cerevisiae*. *BMC Mol. Biol.* **10**, 99 (2009).
109. Mardones, W. *et al.* Rapid selection response to ethanol in *Saccharomyces eubayanus* emulates the domestication process under brewing conditions. *Microb. Biotechnol.* (2021) doi:10.1111/1751-7915.13803.
110. Ibstedt, S. *et al.* Concerted evolution of life stage performances signals recent selection on yeast nitrogen use. *Mol. Biol. Evol.* **32**, 153–161 (2015).

111. Rich, M. S. *et al.* Comprehensive Analysis of the SUL1 Promoter of *Saccharomyces cerevisiae*. *Genetics* **203**, 191–202 (2016).
112. Rest, J. S. *et al.* Nonlinear fitness consequences of variation in expression level of a eukaryotic gene. *Mol. Biol. Evol.* **30**, 448–456 (2013).
113. Bergen, A. C., Olsen, G. M. & Fay, J. C. Divergent MLS1 Promoters Lie on a Fitness Plateau for Gene Expression. *Mol. Biol. Evol.* **33**, 1270–1279 (2016).
114. Alstott, J., Bullmore, E. & Plenz, D. Powerlaw: a Python package for analysis of heavy-tailed distributions. *PLoS One* **9**, e85777 (2014).

## Supplementary Information

### Gigantic Parallel Reporter Assay (GPRA) experimental details

Expression measurements were performed as described in (de Boer *et al.*, 2020) (**Supplementary Fig. 1**). Briefly, a library of ~200,000,000 random 80 bp promoters was cloned in front of a YFP reporter construct within the -160:-80 region of a synthetic promoter scaffold. The promoter scaffold used throughout this study included a distal poly-T tract (5 or more Ts), and a proximal poly-A tract (5 or more As) surrounding the random 80 mers; these features are common in yeast promoters. Furthermore, the scaffold sequences were designed to exclude strong binding sites for TFs. The dual reporter plasmid used is available from AddGene (AddGene:127546) and was derived from the plasmid used by (Sharon *et al.*, 2012). This plasmid contains *URA3*, which we use as a selectable marker, a constitutive RFP (with which to control for extrinsic noise), and the YFP under variable control. Random 80 mers (and designed 80 mer libraries) were cloned into an XhoI site using Gibson assembly. The resulting libraries were transformed into *S. cerevisiae* strains lacking *URA3* using the lithium acetate method (De Boer, 2017), selecting on SD-Ura media, and ensuring that at least 100,000,000 transformants were achieved for the random high-complexity libraries and >100x coverage for designed libraries. Because this is a low copy number CEN plasmid that is segregated like a chromosome during cell division, if a yeast cell is transformed with two different promoters, subsequent cell divisions will ensure with a very high probability that the two plasmids end up in different descendant cells. For random libraries, the strain Y8205 was used, but later experiments including the designed libraries were performed in S288C::*ura3*, which is less auxotrophic. Accordingly, all cases except that in random test dataset (complex media), the models were trained on sequences assayed in one strain of yeast and tested on sequences assayed in another, likely leading to underestimation of the model's performance due to *bona fide* differences between the strains.

Yeast were grown continuously in SD-Ura over the course of two days, and kept in log phase for ~10 generations to allow for reporters to reach equilibrium prior to sorting, diluting the media by 1:4 three times during this period as necessary to keep cells in log phase (OD below 0.8). All cultures were grown in a shaker incubator, at 30°C and approximately 250 RPM. Yeast were harvested by centrifugation, washed once in ice-cold PBS, resuspended in ice-cold PBS, and kept on ice prior to and during sorting. Sorting was performed with a Moflo Astrios (Beckman Coulter) sorting in three sets of 6 bins (all equal width and adjacent) each over the course of ~8 hours, dividing the time equally for the three sets. Cells were sorted by the log ratio of RFP to YFP signal (using mCherry and GFP absorption/emission), which controls for extrinsic sources of variation that affect both reporters (*e.g.*, cell size, plasmid copy number). Once sorted,

cells were kept on ice. Sorted samples were centrifuged to pellet sorted cells, the PBS/sheath fluid aspirated, leaving ~0.5 mL remaining, then the cells resuspended in 1 mL SD-Ura, transferred to a 50mL conical tube containing 9mL media, and the sorting tube washed once with SD-Ura, and transferred to the same conical tube. This produced 18 50 mL tubes each containing ~10 mL of SD-Ura and sorted yeast cells; one per sorting bin. These were allowed to grow for 2-3 days, until all samples reached saturation. Plasmids were isolated using Qiagen spin miniprep kits, as adapted for yeast according to the manufacturer's website (<https://www.qiagen.com/ca/resources/resourcedetail?id=5b59b6b3-f11d-4215-b3f7-995a95875fc0&lang=en>). Nextera adaptors and multiplexing indices were added by PCR, indexed samples were mixed in proportion to the number of cells sorted per bin, and the resulting libraries sequenced paired-end, 76 bp each, using an Illumina Nextseq 500 and 150 cycle kits so that complete coverage of the promoter could be achieved, including overlap in the center.

The sorting bins differed slightly each time FACS was performed for a promoter library. This resulted from the inability of the cell sorter (MoFlo Astrios) to accurately preserve bin configurations on different days and between calibrations. Consequently, the 18 bins were re-assigned for each experiment, and/or laser intensities were adjusted, such that the distribution of RFP:YFP ratios were correctly positioned within the bins. Sorting bins were defined to be uniform in width (expression range) and included the vast majority (>98%) of the distribution and the entirety of the high end of expression, but leaving out the bottom tail of expression. The bottom end of expression tended to be dominated by noise and outliers with abnormally low YFP:RFP ratios. However, the sensitivity at the low end of expression increased in our experiments over time, such that model predictions (in particular for the complex media model) were squished at the low end (e.g. **Extended Data Fig. 1, 3** lower left corners).

The paired reads representing both sides of the promoter sequence were aligned using the overlapping sequence in the middle, constrained to have 40 (+/-15) bp of overlap, and discarding any reads that failed to align well within these constraints. This was not required for the designed libraries. Promoters were aligned to themselves using Bowtie2 (Langmead *et al.*, 2009) to identify clusters of related sequences, merging these clusters and taking the sequence with the most reads as the “true” promoter sequence for each cluster. The designed library reads were aligned to the promoter sequences we ordered using Bowtie2, and only perfect matches were considered in further analysis. Mean expression level for each promoter (as in the processed files) was taken as the average of the bins, weighted by the number of times the promoter was observed in each bin. For the designed libraries (that included all the high-quality test data experiments), we calculated expression for all promoters for which any reads were seen, but used only those



for which we saw at least 100 reads for the analyses described to reduce the amount of measurement error present in the data. For high-complexity random libraries, all promoters were used.

## Biochemical models

The biochemical models were created and used as described previously (de Boer *et al.*, 2020). Code is available on GitHub (<https://github.com/de-Boer-Lab/CRM2.0>). Briefly, the models are trained using the “makeThermodynamicEnhancosomeModel.py” program within the <https://github.com/de-Boer-Lab/CRM2.0/blob/master/usefulScripts/makeProgressiveBiochemicalModels.bat> script (using the “110 - eb” parameters which describe the sequence length (110) and the expected binding TF model (-eb)). Training happens in 5 stages, with each subsequent stage restoring the parameters learned in the previous step before continuing training, and optimizing the noted new parameters as well as all others that were previously learned: (1) potentiation and activity parameters are learned, after having initialized the motifs to known motifs for each TF, and TF concentrations initialized to the min  $K_d$  possible with each motif (corresponding to 50% occupancy of a perfect binding site); (2) concentration parameters are optimized; (3) motif models are optimized; (4) TF binding/activity limits are introduced and optimized; and (5) position-specific activities are introduced and optimized.

Each training round is performed with a full epoch of the training data (5 epochs total). Inference is performed using the “predictThermodynamicEnhancosomeModel.py” program. In order to get regulatory strength for a TF, the “-dotf” parameter was used, and inference run again. This parameter sets the concentration parameter of the indicated TF to 0, and then predicts expression. An example for how to use these parameters and programs to calculate regulatory complexity is included here: <https://github.com/de-Boer-Lab/CRM2.0/tree/master/usefulScripts>. For all analyses, the biochemical models using position-specific activities were used with the exception of the biochemical model-derived ECC, where the non-positional model was used, because the position-specific activity parameters we had previously found are partly dependent on the surrounding sequence context (de Boer *et al.*, 2020). The decrease in % error ( $100\% \times (1-r^2)$ ), that is, the fraction of variance unexplained relative to the biochemical model is around  $\sim 45\%$  ( $((0.96^2 - 0.926^2) / (1 - 0.926^2))$  (positional biochemical model) for the Native test data. The biochemical models were used in sections where we required model interpretability (**Fig. 2d**), but the deep learning models were used elsewhere, since the biochemical models are slower than the deep learning models to run inference on and have lower predictive performance on the test data.

## ECC calculation details and considerations

The ECC depends on both simulated and natural variation in promoter sequences. The natural variation in promoters is not independently sampled, since promoters from closely related strains often have identical sequences. Consequently, even when there are 1,011 orthologous promoters for each gene in the 1,011 whole yeast genomes dataset, there will typically be many fewer unique promoter sequences. Meanwhile, each sequence in the simulated variation is sampled independently (to increase robustness of the estimation of the null expectation), so, here, there are often 1,011 unique promoter sequences. The simulated variation was generated by placing random mutations within the gene’s promoter consensus (the most abundant base at each position in the orthologous set), while preserving the Hamming distance distribution observed in the natural sequences (**Methods**). Despite these sets each having the same Hamming distance distribution relative to the consensus, the standard deviation (SD) calculated from N independently sampled sequences (as in the simulation) is biased towards being greater than that for N dependently sampled sequences (as in evolution), resulting in the raw ECC values being biased in favor of “conservation” as a result of a statistical bias rather than due to selection.

To demonstrate that this bias is not evolutionary in nature we calculated a “mock” ECC where both the numerator and denominator represent simulated variation. In the mock ECC, the sequences in the numerator are sampled independently and match the Hamming distance distribution of the natural variation (as in the standard ECC), but the denominator (normally the natural variation) is sampled in a way that matches *both* the Hamming distance distribution *and* the number of unique sequences at each Hamming distance (relative to the natural variation). Despite both sets of sequences being randomly sampled and having matched Hamming distance distributions, the mock ECC is slightly positively biased (**Supplementary Fig. 2a**), highlighting the need for a correction factor. Consequently, we used the median of these mock ECCs  $\left(\log_2\left(\frac{\sigma_{C_i}}{\sigma_{C_i}}\right)\right)$  as the correction factor.

While it is theoretically better to have gene-specific correction factors, *these* are much more computationally intensive to calculate and provide little benefit in practice. To generate gene-specific correction factors, we need to make many instances of simulated variation for each gene, estimate the gene-specific bias, and use it to correct the observed ECC. Doing this with 1,111 simulations for each gene showed that there was little difference in the resulting ECC values, compared to a global correction factor (**Supplementary Fig. 2b**). Given the computational intensity of this approach (which, after optimization, still takes several days to run) and low practical utility, we favored the approach with a global correction factor. We do provide the gene-specific corrected ECCs in **Supplementary Table 1**.

The substitution rate in the genome is not uniform, but we use a uniform substitution rate when calculating the ECC. To test for the impact of this choice, we re-calculated the ECC using the substitution rates observed in the 1,011 yeast genomes promoters and found that the ECCs were largely concordant (**Supplementary Fig. 2c**). Since the mutations we observe in promoters are themselves biased (having survived selection), both approaches yield similar ECC values, and it is much easier to use a uniform base substitution rate, we use the uniform substitution rate ECC throughout the study.

Finally, we note that our approach for computing the ECC assumes that the relative effects of mutations within a sequence are similar regardless of the surrounding sequence context.

## Comparison of ECC to RNA-seq expression

We examined the robustness of our finding that ECC distributions differ significantly between genes with conserved and divergent expression (by RNA-seq) to the threshold we chose to define expression conservation. To this end, we performed the Wilcoxon rank sum test analysis across a range of thresholds for each dataset. Both the *Saccharomyces* and Ascomycota results were significant ( $P < 0.05$ ) at all thresholds, and much more significant ( $p < 10^{-5}$ ) at a threshold of 10% and above (**Supplementary Fig. 3a-c**).

For mammals, we used the threshold of 25% applied in the original publication (Chen *et al.*, 2019). In addition, we performed the Wilcoxon rank sum test analysis across a range of thresholds and found that the results were similarly significant for the full range of thresholds bar one (5%, the lowest threshold; **Supplementary Fig. 3d**). The null hypothesis could not be rejected at the 5% threshold, given the smaller number of yeast gene one-to-one orthologs in mammals in both the expression conservation classes.

In principle, the ECC can be calculated across orthologous regulatory sequences from many different species (as opposed to individuals within a species, as we did here), but we advise caution if doing so. The ECC assumes that the function relating sequence to gene expression is the same across the orthologous sequences being compared. Since regulatory sequences evolve much faster than the regulators themselves (Weirauch and Hughes, 2010), this assumption is likely a reasonable approximation within a species, but as evolutionary distances increase, regulators will diverge, gradually eroding this assumption. An alternative is to use gene orthology to infer the extent of expression conservation in one species using

ECCs calculated in another species (**Extended Data Fig. 4b**). However, such relations would extend only to well-mapped orthologs.

## Benchmarking of sequence-to-expression models

We examined different neural network architectures for their ability to predict expression when trained on our data. We compared our transformer model to three model architectures from existing literature (Agarwal *et al.*, 2020) on gene expression prediction models: DeepAtt (Li *et al.*, 2020), DeepSEA (Zhou and Troyanskaya, 2015), and DanQ (Quang and Xie, 2016). (We focus here on comparison to the transformer model, as the convolutional model was not used for some of the compared tasks, such as calculation of the ECC. However, equivalent comparisons can be made with the convolutional using the code shared) Although these models differ from our own and from each other, we adopted each of the model architectures for our application to the best of our ability using the source code (<https://github.com/jiawei6636/Bioinfor-DeepATT>) from each original publication (the adopted model architecture implementation can be found on our GitHub repo at: [https://github.com/1edv/evolution/tree/master/manuscript\\_code/model/benchmarking\\_models](https://github.com/1edv/evolution/tree/master/manuscript_code/model/benchmarking_models)) for the purpose of this benchmarking analysis. The precise details of the benchmarking architectures can be found in the code, and are described below. Note, that the input and output layers (which are the same for each model) are omitted from the lists below.

### 1) DeepATT :

- Convolution (filters=256, kernel\_size=30)
- MaxPool (pool\_size = 3, strides = 3)
- Dropout (0.2 probability)
- BiDirectional LSTM (16 units)
- MultiHeadAttention
- Dropout (0.2 probability)
- Dense (16 units)
- Dense (16 units)

### 2) DeepSEA :

- Convolution (filters=320, kernel\_size=8)
- MaxPool (pool\_size = 3, strides = 3)

- Dropout (0.2 probability)
- Convolution ( filters=480, kernel\_size=8)
- MaxPool (pool\_size = 3, strides = 3)
- Dropout (0.5 probability )
- Dense (64 units)
- Dense (64 units)

3) DanQ :

- Convolution (filters=320, kernel\_size=26)
- MaxPool (pool\_size = 3, strides = 3)
- Dropout (0.2 probability)
- BiDirectional LSTM (320 units)
- Dropout (0.5 probability)
- Dense (64 units)
- Dense (64 units)

Next, we trained each of these adapted models using the same training data (in complex media) as the original convolutional and transformer model, and tested each of the model's predictive power on a set of high-quality native DNA sequences measured in our system. We found that our transformer model outperformed the other three architectures on these data (**Supplementary Fig. 4a**), as expected given that these other approaches were designed for other purposes.

We also used each of these models to calculate the ECC, finding that the resulting ECC values are highly correlated to the ECCs predicted by the transformer model (**Supplementary Fig. 4b-d**). This shows that our framework leads to equivalent biological conclusions when used with model architectures that have overall comparable predictive performance.

To rule out the possibility that the transformer model's increased performance results from learning of technical biases, we compared the transformer model's ECC to an ECC calculated using the interpretable biochemical model (de Boer *et al.*, 2020), also trained using GPRA data, which, with a single convolutional layer and many fewer parameters, is presumably less able to capture technical biases. Here too, we found that the ECCs are highly similar between the two models (**Supplementary Fig. 5g**). Finally, we found that the ECC values computed using the transformer model are better at predicting expression conservation as measured by RNA-seq across the range of possible thresholds considered (**Supplementary Fig. 5h**).

## Ablation analysis of the sequence-to-expression transformer model

The transformer model was motivated by several intuitions aimed to help it leverage known aspects of *cis*-regulation(Weirauch *et al.*, 2013; Brodsky *et al.*, 2020), but which may or may not be explicitly captured. The first convolutional block with three layers, was motivated by the idea to identify sites that are important for computing the expression target, and could be analogous to a TF scanning the length of the sequence for binding sites. The first layer was aimed towards an abstract representation of first order TF-sequence interactions by operating with convolutional kernels on the sequence in the forward and reverse strands separately to generate strand-specific features (each individual kernel in the first layer can be thought of as possibly learning the motif of one TF, or a combined representation of the motifs)(Alipanahi *et al.*, 2015; Zhou and Troyanskaya, 2015; Shrikumar, Greenside and Kundaje, 2017; Quang and Xie, 2019) and we designed the width of the first convolutional layer (30 bp) to be sufficient to capture the largest TF motifs known in yeast(de Boer and Hughes, 2012); the second was aimed towards capturing interactions between strands, by using a 2D convolution (implemented using the *tf.keras.layers.Conv2D* layer, and convolving along the sequence dimension) on the combined features from the individual strands; and the third layer was aimed towards capturing higher order interactions, such as TF-TF cooperativity. We zero-pad the convolution blocks to allow the convolutional filters to detect motif instances near the edges of the input sequence. The second block was motivated by an analogy to combining the biochemical activities of multiple bound TFs and accounting for their positional activities. Its transformer-encoder with a multi-head self-attention module(Vaswani *et al.*, 2017) could capture relations between features extracted by the convolutional block at different positions in the sequence, by attending to them simultaneously using a scaled dot product attention function. This could be analogous to the model learning ‘where to look’ within the sequence. Then, a bidirectional Long Short-Term Memory (LSTM) layer in this block was motivated by the idea of capturing long range interactions between the sequence regions. Finally, a multi-layer perceptron block was motivated by the idea of capturing cellular operations that occur after TFs are recruited to the promoter sequence, by pooling all the features extracted from the sequence through the previous layers and learning a scaling function that transforms these abstract feature representations of biomolecular interactions into an expression estimate. While these were our motivations in architecting the model, because our focus was predictive ability and not interpretability of regulatory mechanisms, we do not know if the model in fact captured these relations in this way.

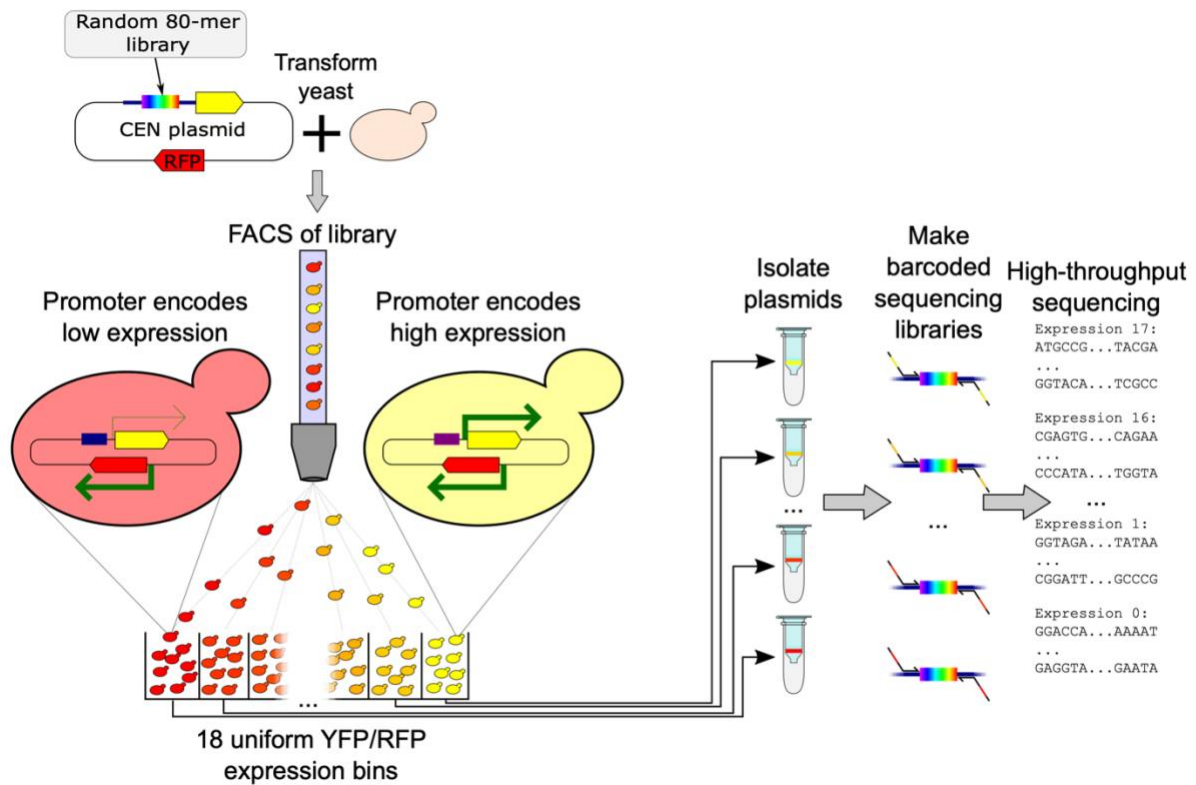
In order to determine whether any of the transformer model’s layers were superfluous, we conducted an ablation study. For each ablation experiment, we initialized a new model from scratch after removing the

ablated layer individually from the original transformer model architecture, while retaining every other component of the original transformer model. Then, we trained this new model using the same training data (in complex media) as the original transformer model, and tested the resulting models on the high-quality random DNA test data. We found that each layer has non-trivial individual contributions to our predictions, with the full model performing better than any of the ablated models (**Supplementary Fig. 6**).

## Expression distribution at the robustness cleft and the malleable archetype

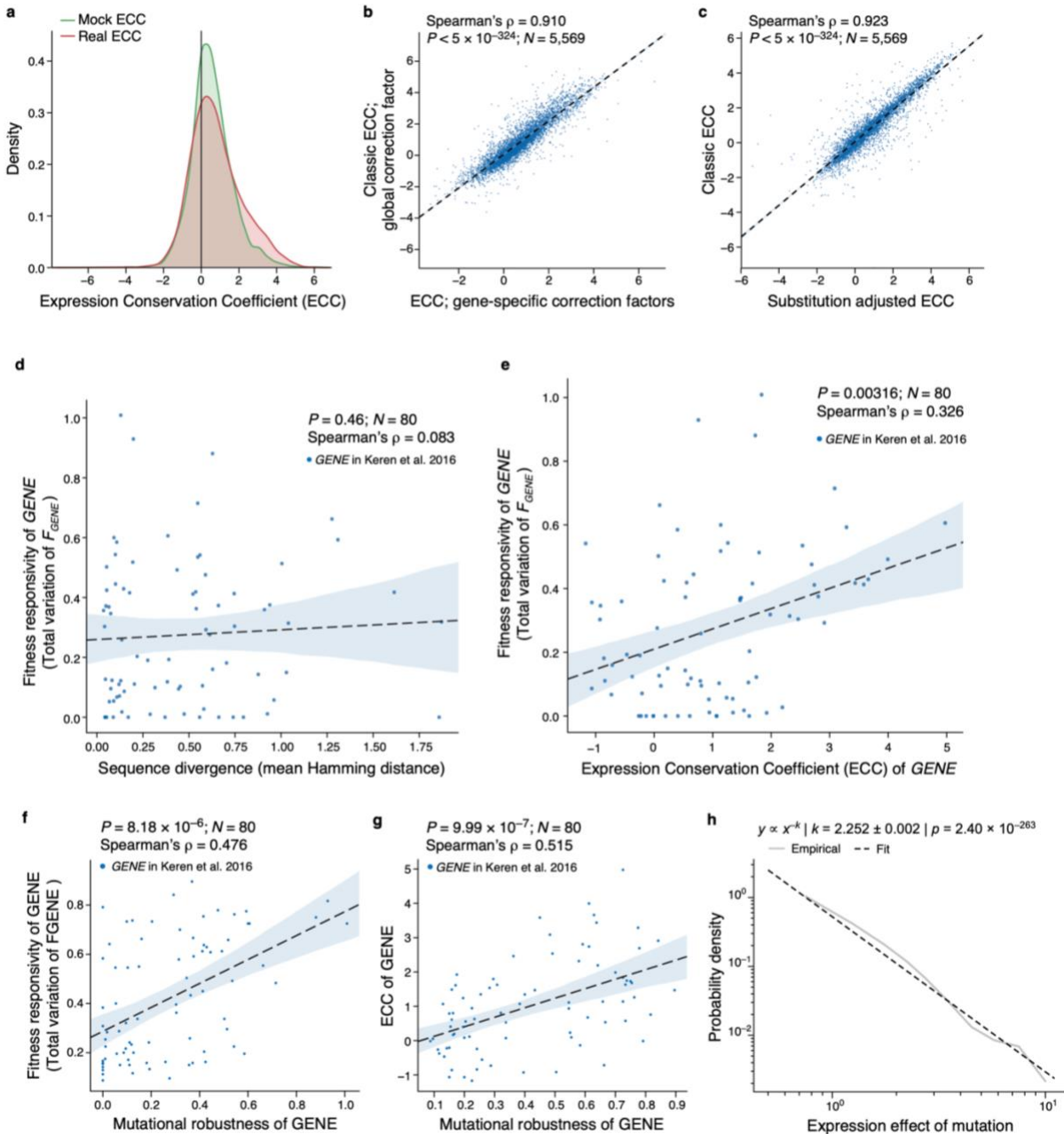
While our observation that sequences with intermediate expression levels are more likely to be near the malleable archetype ( $A_{\text{malleable}}$ ) and depleted near the robustness cleft (**Fig. 4d**), could in theory result from a saturation artifact of our reporter construct, our ratiometric sorting strategy allowed us to detect saturation and none was observed. Instead, the robustness cleft could reflect sequences at the stable extremes of one or more activation steps of gene expression (e.g. near 100% or 0% nucleosome occupied), while the malleable archetype could reflect instability around the inflection points.

## Supplementary Figures



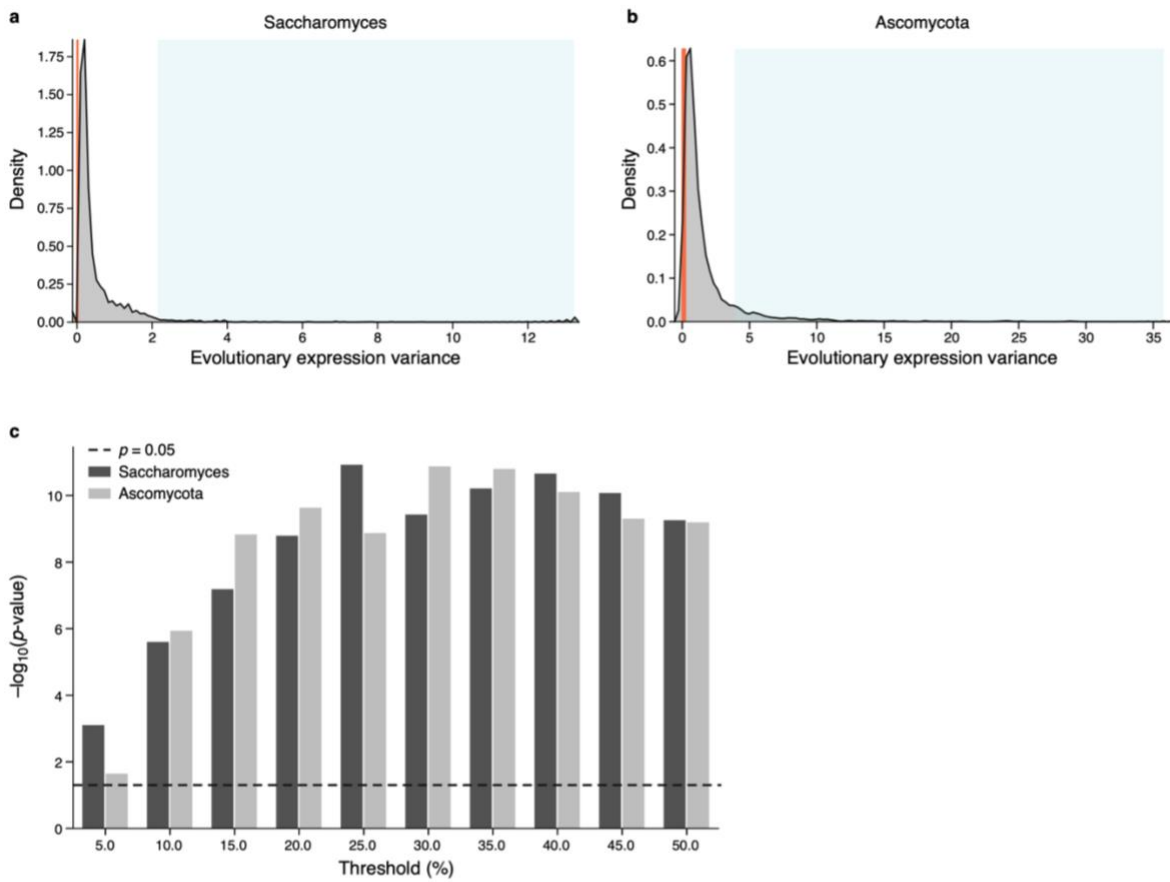
**Supplementary Fig. 1 | GPR experiment overview.** Yeast are transformed with a library of random 80 bp sequences driving YFP expression, the cells recovered and selected for successful transformants, and grown in the target media in log phase. Yeast are then sorted by the ratio of YFP to RFP into 18 different uniform expression bins. Yeast are then recovered in selection media (SD-Ura), plasmids isolated, sequencing libraries created, and the promoters in each expression bin sequenced with high-throughput sequencing.





**Supplementary Fig. 2** | **a**, Comparison of raw ECC distributions for natural variation (red) and matched simulated variation (green, “mock ECC”). Both are biased towards having an ECC above 0. **b**, Comparison of ECCs with global correction (x axis) and gene-specific correction factors (y axis). **c**, ECC with uniform substitutions (y axis) is highly correlated to the ECC computed using the observed substitution rate (x axis). **d**, **e**, Fitness responsivity is not associated with simple sequence diversity, but is associated with ECC. Fitness responsivity (y axes) and mean Hamming distance (**d**, x axis) or ECC (**e**, x axis) for each of 80 genes (points). **f**, **g**, Genes whose expression changes have stronger effects on organismal fitness have mutationally robust regulatory sequences. Mutational robustness (x axes) and fitness responsivity (**f**, y axis) or ECC (**g**; y axis) for each of 80 genes (points) for which the expression-to-fitness curves were quantified (Keren *et al.*, 2016). (**b-g**) Spearman’s  $\rho$  and associated two-tailed p-values are shown. The light blue error bands represent the respective 95% confidence intervals. **h**, Mutational effects

follow a power law distribution. Probability density ( $y$  axis) and expression effect of mutation (magnitude) ( $x$  axis) plotted on log-log axes (solid line) alongside the goodness of fit (dash line) of the power law distribution.

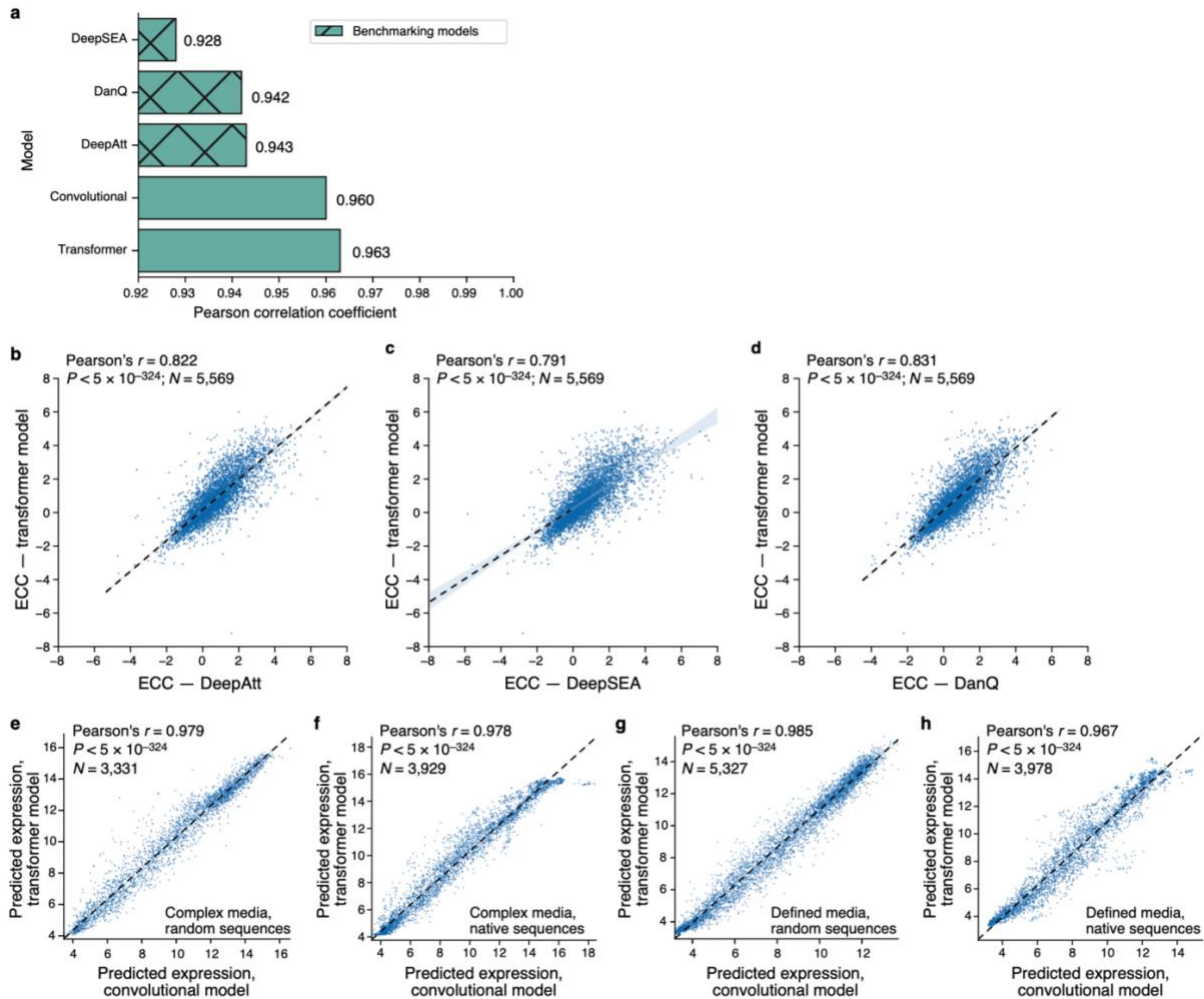


**d**

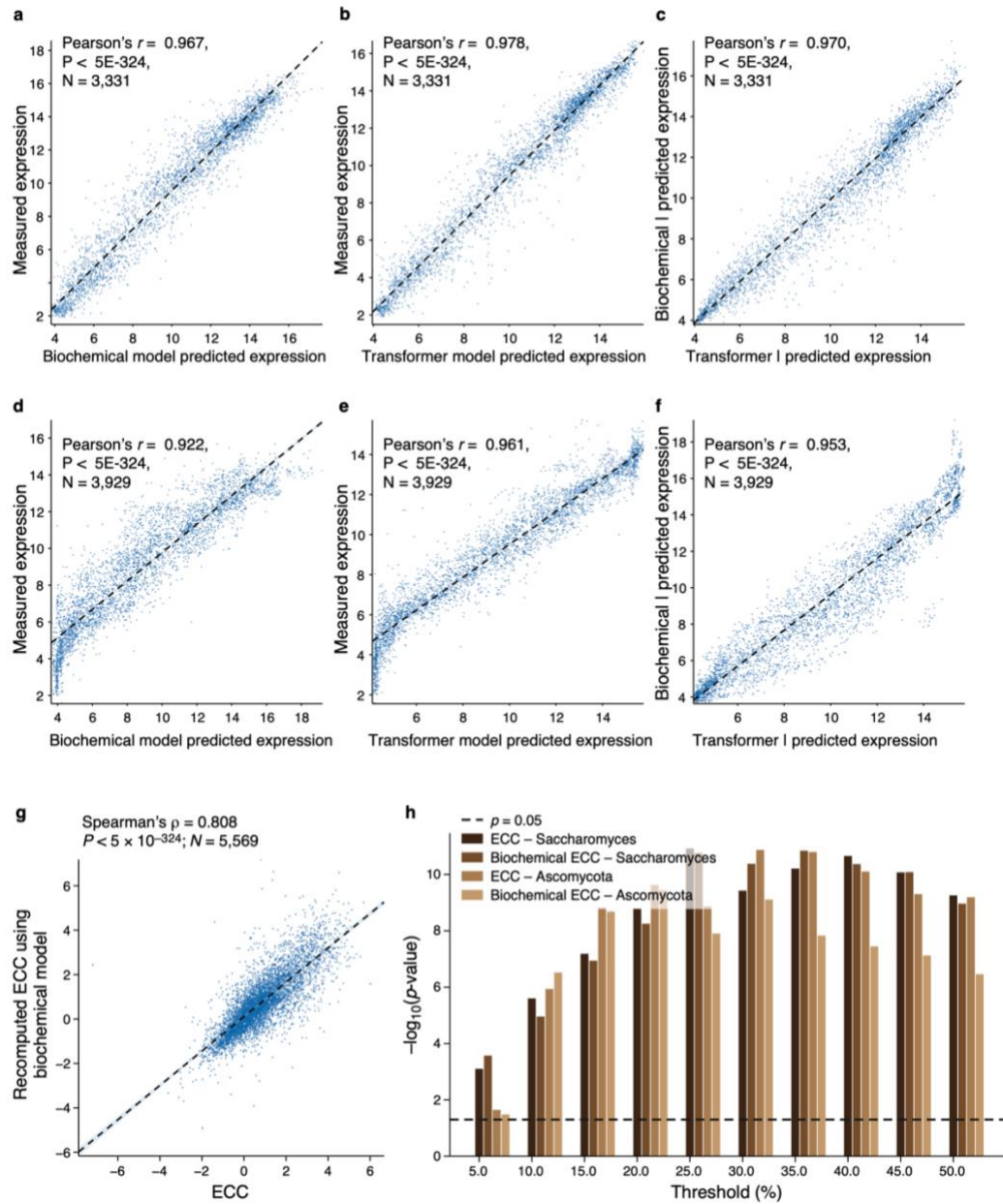
Threshold (%)	<i>n. genes (conserved)</i>	<i>n. genes (not-conserved)</i>	Mammalian p-value
5.0	144	75	0.132141
10.0	226	137	0.001095
15.0	291	208	0.000609
20.0	347	263	0.000897
25.0	333	288	0.000107
30.0	300	274	0.000040
35.0	261	246	0.00019
40.0	208	215	0.007662
45.0	173	185	0.009174
50.0	144	155	0.001364

**Supplementary Fig. 3 | a,b**, Expression variance (by RNA-seq) for *Saccharomyces* (a) and Ascomycota (b). Green boxes: genes called as divergent; orange: genes called as conserved by the thresholds in this study (as in **Extended Data Fig. 4b**). **c**, Sensitivity of ECC enrichment significance (Wilcoxon rank sum test  $-\log_{10}(P\text{-values})$ ; y axis) to “conserved” vs. “divergent” thresholds (x axis) for Ascomycota (light gray) and *Saccharomyces* (dark gray).  $P=0.05$ : dashed line. **d**, Sensitivity of ECC

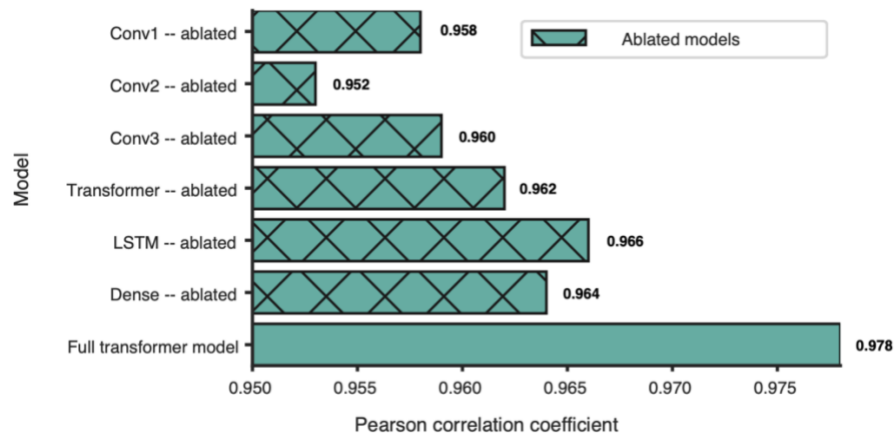
enrichment significance (Wilcoxon rank sum test, “Mammalian p-value”) to “conserved” vs. “divergent” thresholds (“Threshold (%)”) in mammals. The columns display the number of genes determined to be in each class at each threshold.



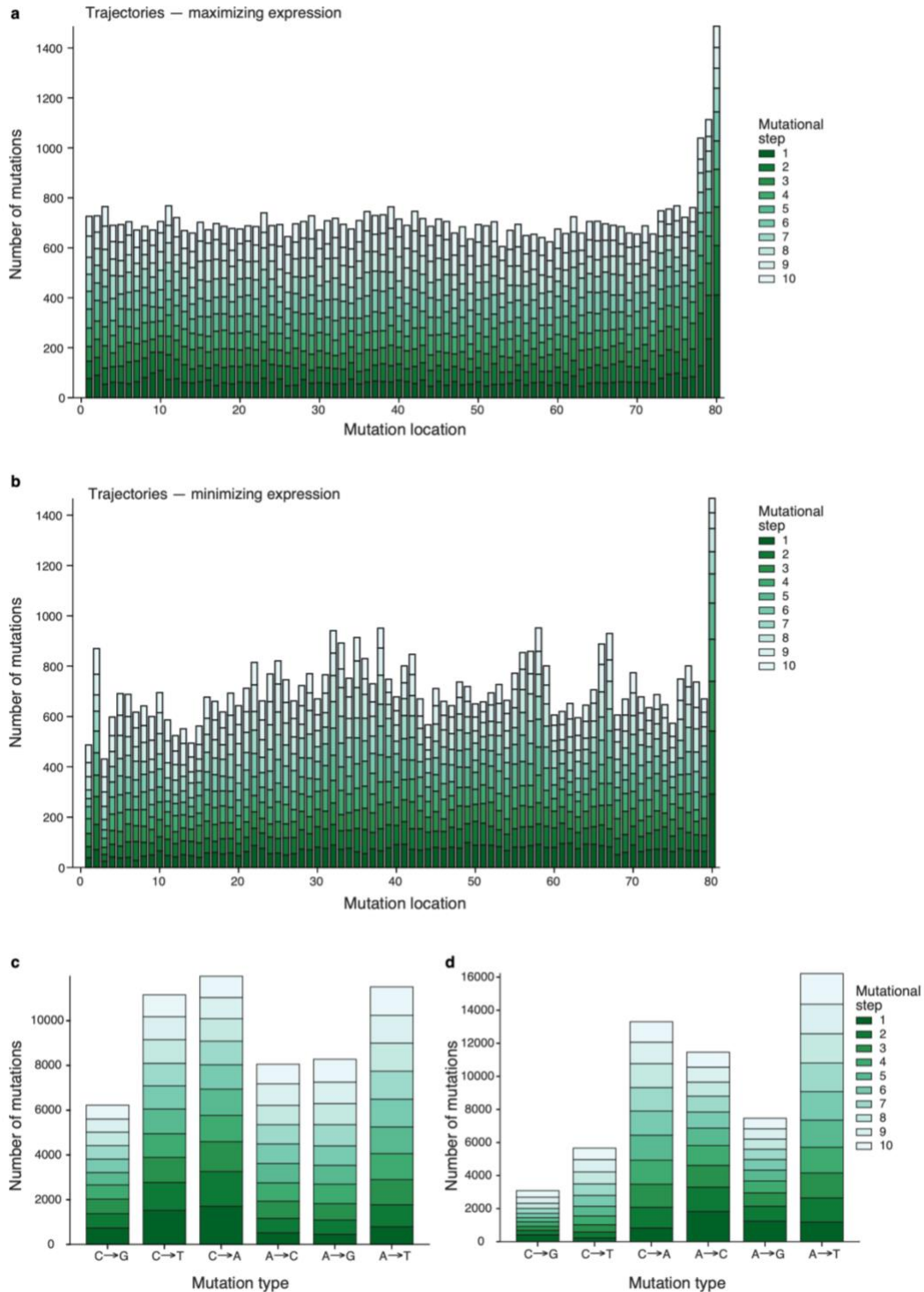
**Supplementary Fig. 4 | a**, Benchmarking of performance against existing neural network architectures. Pearson correlation coefficient between model predictions and test data ( $x$  axis) for four model ( $y$  axis). All models were trained on the same training dataset, and tested on the same set of native promoter test sequences in complex media. While all approaches performed reasonably well, the transformer model architecture used in this paper out-performed the others on the native test sequence dataset. **b-d**, Comparison of ECC calculated with our model ( $y$  axis) and with (**b**) DeepAtt, (**c**) DeepSEA and (**d**) DanQ ( $x$  axis). In each case, the ECC predictions are highly correlated between each approach and our model. (Outliers not shown for the panel (**c**) to maintain scaling and visibility; Pearson's  $r$  was computed using all of the data including outliers.). **e-h**, The convolutional and transformer models have highly correlated predictions. Predicted expression from the convolutional ( $x$  axis) and transformer ( $y$  axis) models in complex (**e-f**) and defined (**g-h**) media for random (**e-g**) and native (**f-h**) test datasets. (**b-h**) Pearson's  $r$  and associated two-tailed  $p$ -values are shown.



**Supplementary Fig. 5 | Comparison of the biochemical and transformer models.** Measured and predicted expression in complex media for (a-c) random test data as, and (d-f) native test data. (a,b,d,e) Measured (y-axes) and predicted (x-axes) expression, for (a,d) biochemical and (b,e) transformer models. (c,f) transformer (x-axes) and biochemical (y-axes) model predictions. (a-f) Pearson's  $r$  and associated two-tailed p-values are shown. g,h, The transformer model outperforms the biochemical model in differentiating expression conservation status. g, Comparison of ECCs calculated for each gene (points) for the transformer model (x axis) versus the biochemical model (y axis). Spearman's  $\rho$  and associated two-tailed p-values are shown. h, Significance (y axis) of rank sum statistics for how well ECCs calculated with each method separates conserved versus not conserved genes across *Saccharomyces* (dark brown) and Ascomycota (light brown).



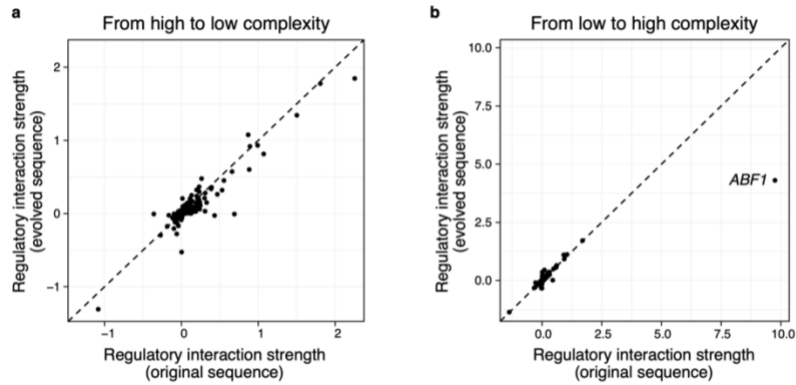
**Supplementary Fig. 6** | Each layer individually contributes to model performance. Performance ( $x$  axis, Pearson's  $r$  between the model predictions and random test data) of the transformer model variants ( $y$  axis) with each layer individually ablated, and the full transformer model (bottom). The full transformer model outperforms all other versions with any model component ablated. The two-tailed  $p$ -value corresponding to each performance metric shown is  $< 5 \cdot 10^{-234}$ .



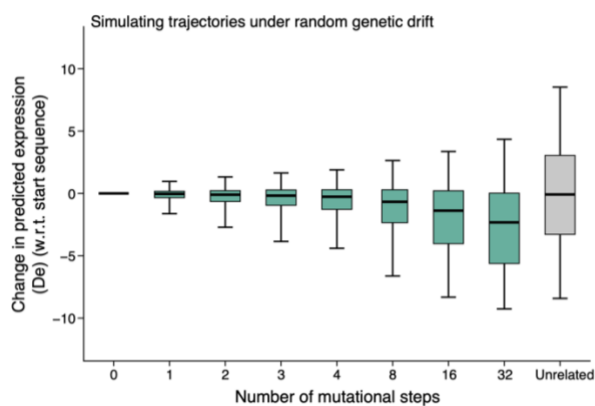
**Supplementary Fig. 7 | Sequences took diverse paths to evolve extreme expression. a-b,** The number (*y* axis) of mutations across each promoter position (*x* axis; -160 to -80 region) for trajectories under the SSWM regime starting with native promoter sequences when **(a)** maximizing or **(b)** minimizing expression in defined media using the convolutional model. Some of the observed bias to TSS-proximal mutations may be related to prior observations of proximal repressor activity bias (de Boer *et al.*, 2020). **c,d,** The number (*y* axis) of mutations of each type (*x* axis) for trajectories under the SSWM regime starting with native



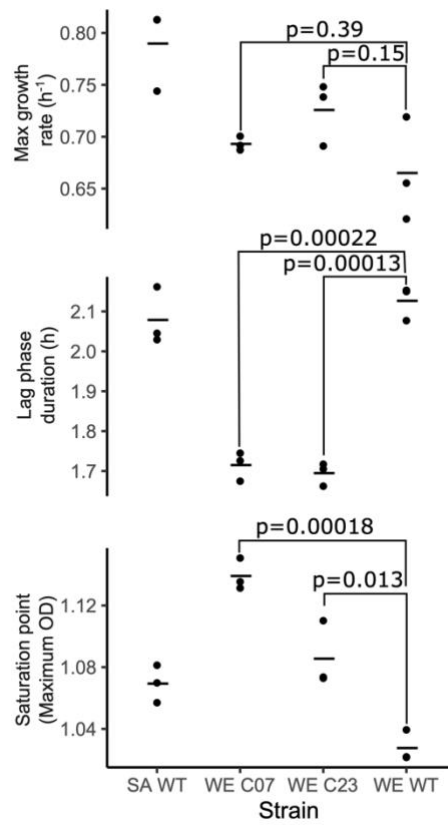
promoter sequences when **(c)** maximizing or **(d)** minimizing expression in defined media. Colors represent the mutational step (1-10).



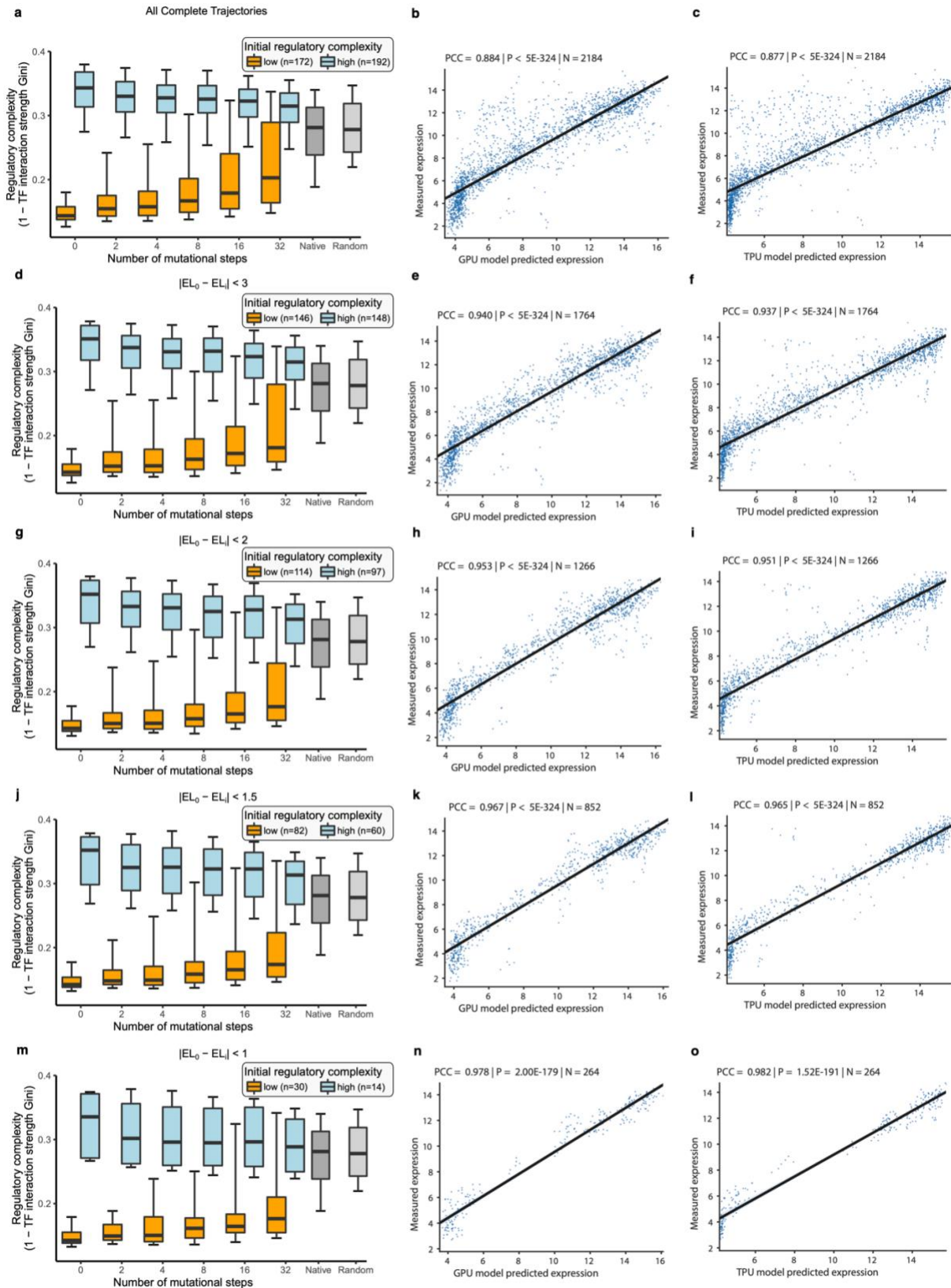
**Supplementary Fig. 8 | a,b,** Examples of regulatory complexity changes under stabilizing selection. TF regulatory interaction strengths for original ( $x$  axes) and evolved ( $y$  axes) sequences after 32 neutral (expression maintaining) mutations for each TF (points) for -160:-80 promoter regions for (a) *YDR476C*, whose regulatory complexity was high and decreased (from 0.3 to 0.25), and (b) *AIF1*, whose complexity was low (dominated by the TF Abf1p) and increased (from 0.14 to 0.21). Both have approximately the same predicted expression levels (13.7 and 14.3 respectively).



**Supplementary Fig. 9** | Predicted expression divergence under random genetic drift. Distribution of the change in predicted expression (y axis) for native yeast promoter sequences (n=5,720) at each mutational step (x axis) for trajectories simulated under random mutational drift using the transformer model. Silver bar: differences in expression between unrelated sequences. Expression decreases with increasing mutation number because the average expression of the starting set of native sequences is greater than for random DNA, and so including random mutations are more likely to decrease expression than increase it. Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentiles.

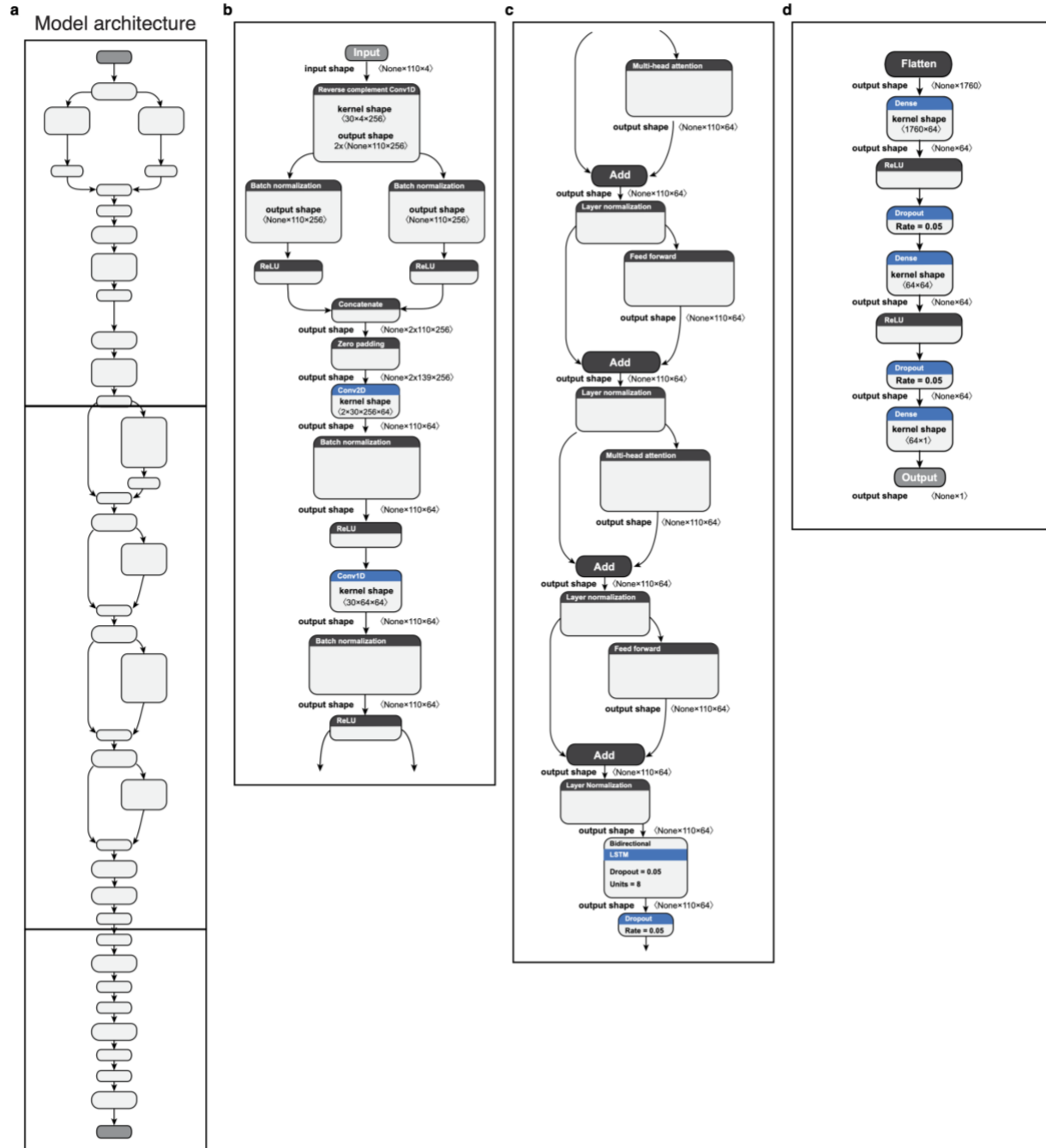


**Supplementary Fig. 10** | Growth phenotypes of *CDC36* promoter mutant strains. Maximum growth (*y* axis, top), duration of lag phase (*y* axis, middle) and saturation of growth (*y* axis, bottom) for two WT strains and two engineered strains (*x* axis). Bars: means, dots: replicate measurements. P-values: Student's *t*-test; two-sided, unpaired, equal variance.  $n=3$  replicates/strain.

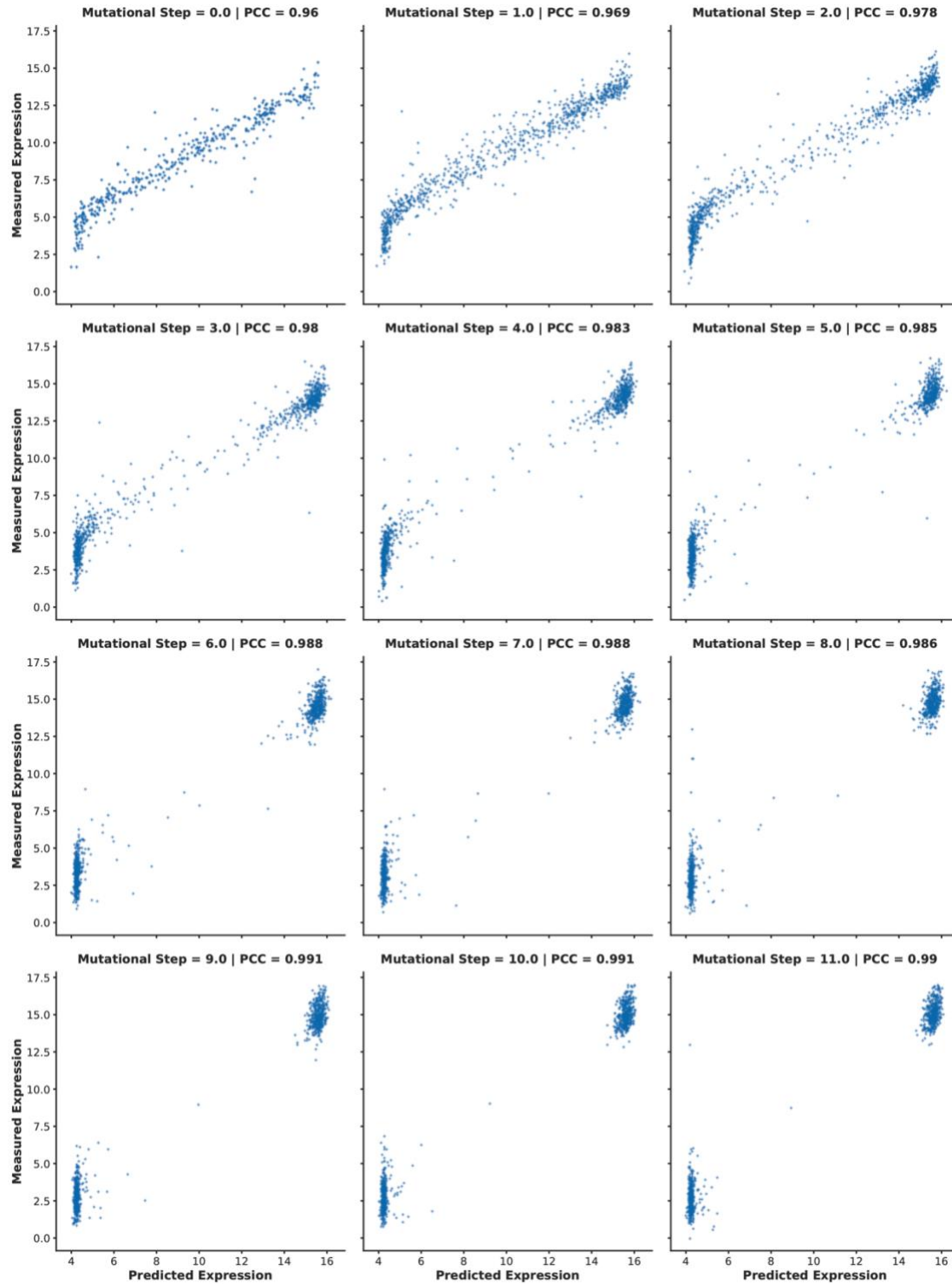


**Supplementary Figure 11: Robustness of moderation of regulatory complexity to the degree of stabilizing selection.** (a,d,g,j,m) Distributions of regulatory complexity (y-axes) for sets of sequences with initial high (light blue) and low (orange)

regulatory complexity, and evolved sequences at different mutation steps, with native and random sequences shown for reference (dark and light gray respectively). Here,  $n$  is the number of trajectories included. All evolved sequences were designed to mimic stabilizing selection by requiring that expression changes by no more than 0.5 expression units relative to the original using the GPU model. Also shown are the measured ( $y$ -axes) and model predicted ( $x$ -axes) expression levels for the convolutional (**b,e,h,k,n**) and transformer (**c,f,i,l,o**) models. Results are shown for all complete experimental trajectories (**a-c**), or when including only trajectories where no evolved sequences had measured or transformer model-predicted expression that differed from the measured expression of the original sequence by more than 3 (**d-f**), 2 (**g-i**), 1.5 (**j-l**) or 1 (**m-o**) expression units. All data are for complex media (YPD). (**a,d,g,j,m**) Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. (**b,c,e,f,h,i,k,l,n,o**) Pearson's  $r$  and associated two-tailed  $p$ -values are shown.

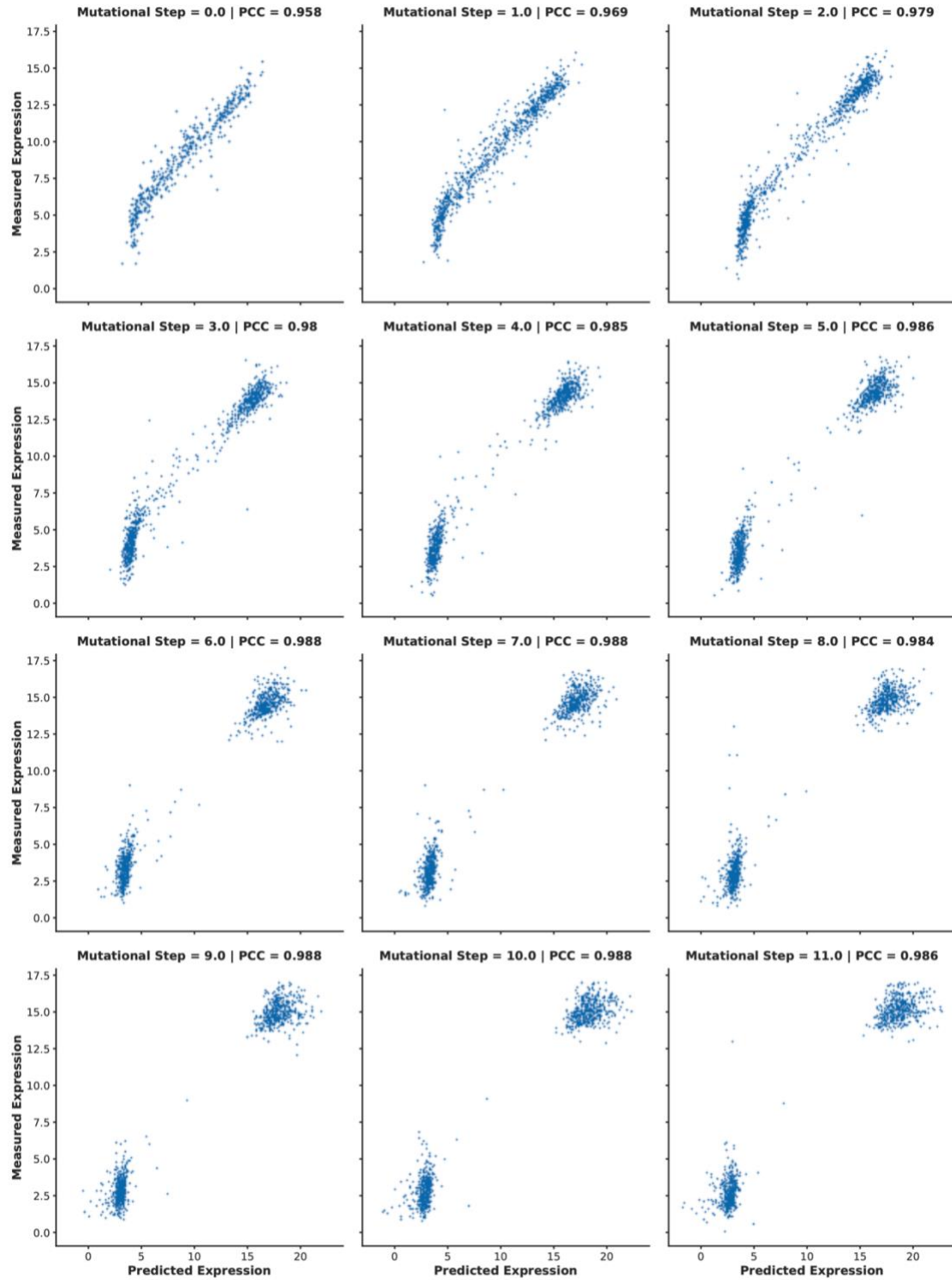


**Supplementary Fig. 12 | The deep transformer neural network architecture for the sequence-to-expression model. a,** Model architecture with three blocks (horizontal lines) and multiple layers (boxes). **b-d.** Expanded architecture (**Methods**) for the convolutional (**b**), transformer encoder (**c**) and multi-layer perceptron (**d**) blocks in our transformer model.

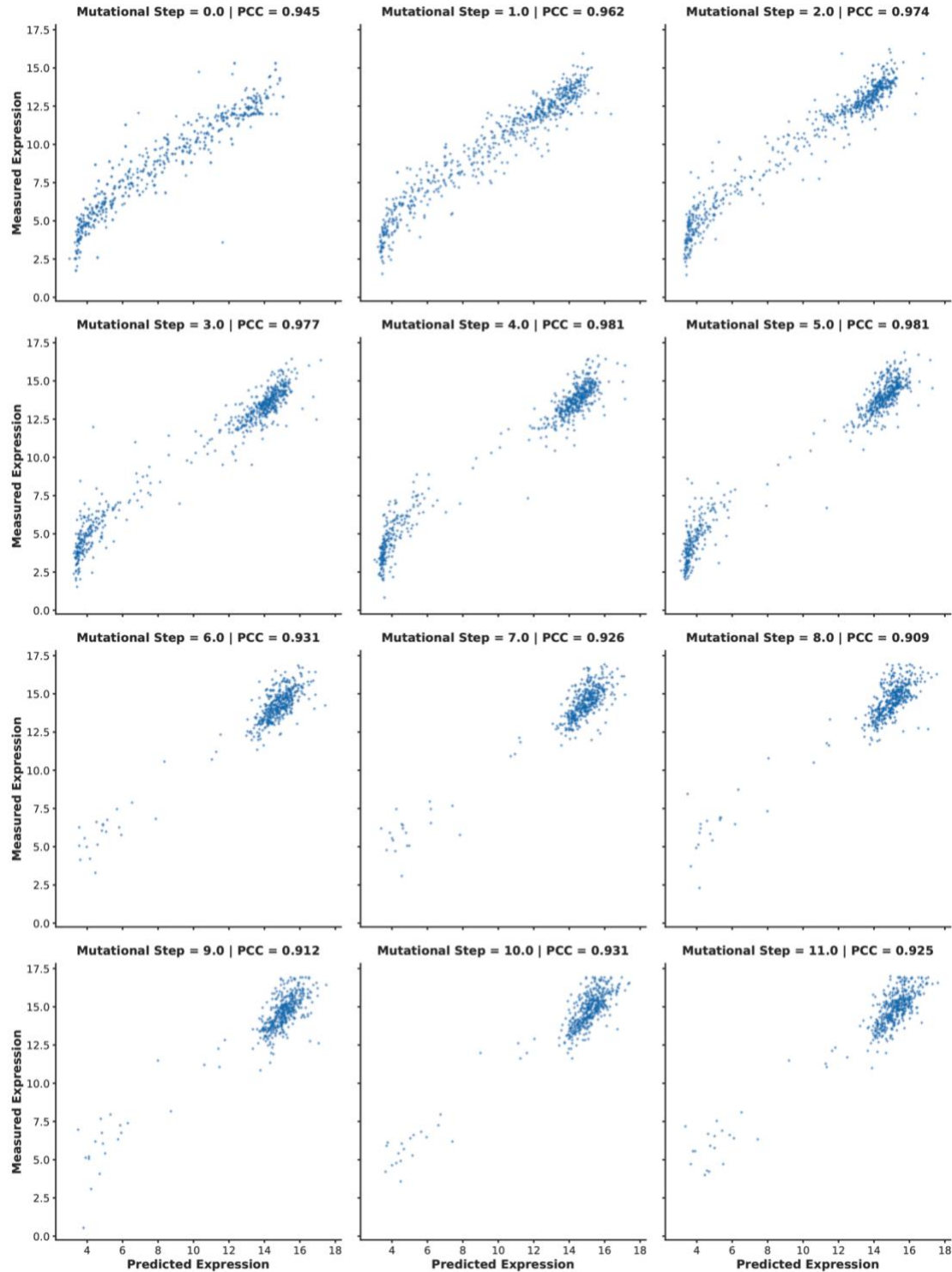


**Supplementary Fig. 13 | Comparison between predictions and measurements of expression at each mutational step in complex media.** Transformer model predicted (*x*-axes) and measured (*y*-axes) expression for each mutational step (plots) for trajectories in **Fig. 2g** ( $n=10,322$  sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. The two-tailed *p*-value corresponding to the performance metric shown in each panel is  $< 5 \times 10^{-234}$ .

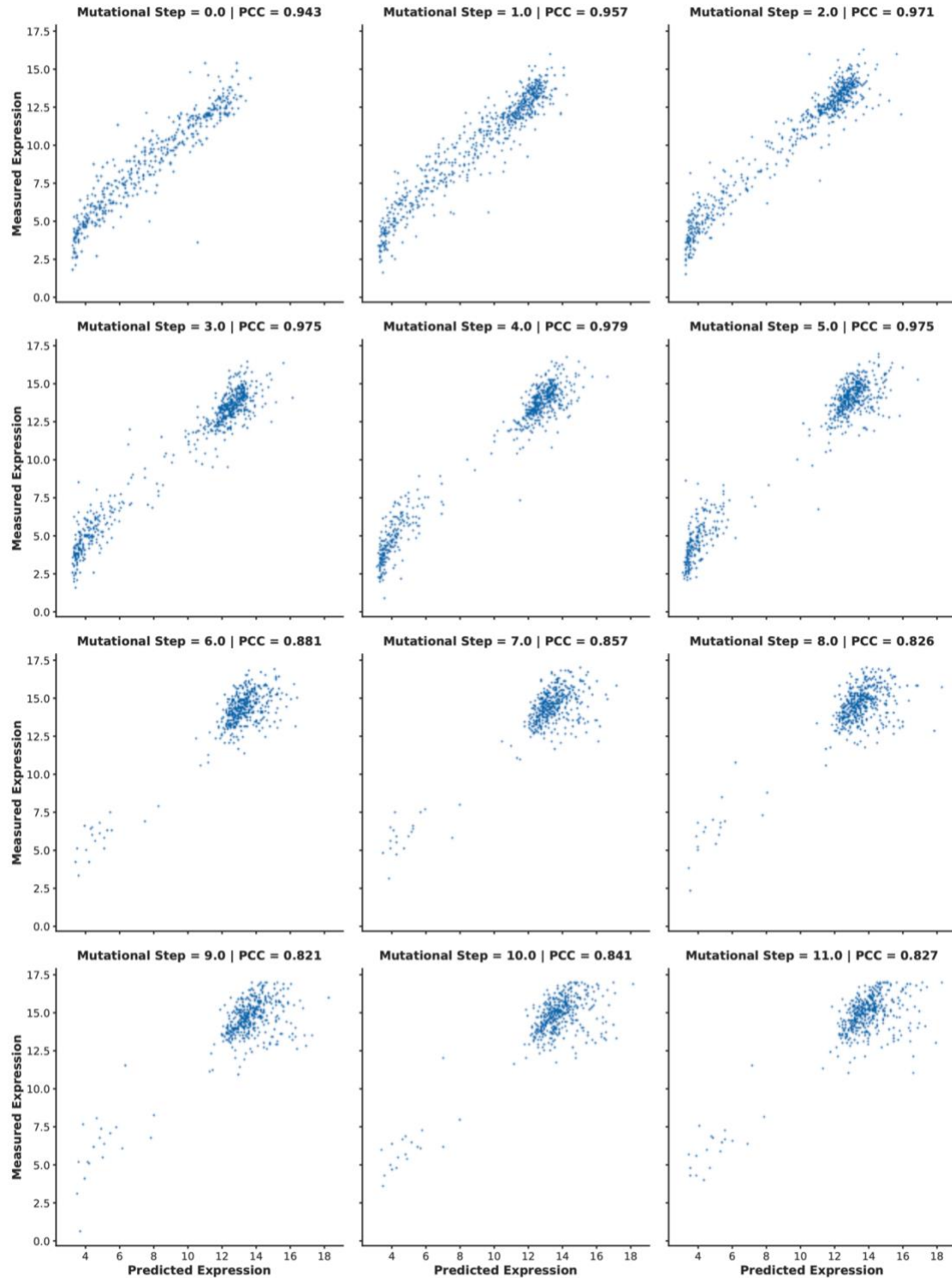




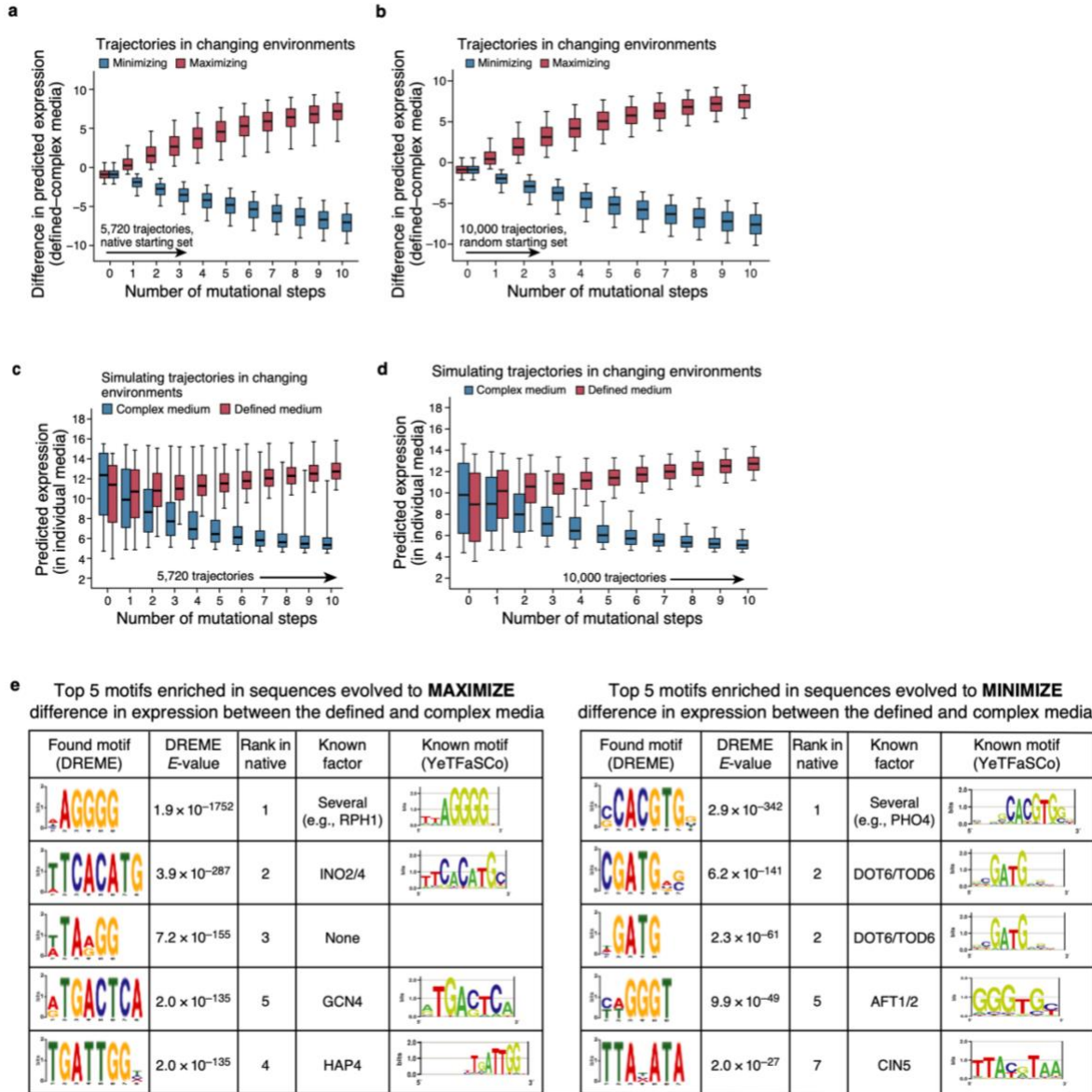
**Supplementary Fig. 14 | Comparison between predictions and measurements of expression at each mutational step in complex media.** Convolutional model predicted ( $x$ -axes) and measured ( $y$ -axes) expression for each mutational step (plots) for trajectories in **Fig. 2g** ( $n=10,322$  sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. The two-tailed  $p$ -value corresponding to the performance metric shown in each panel is  $< 5 \cdot 10^{-234}$ .



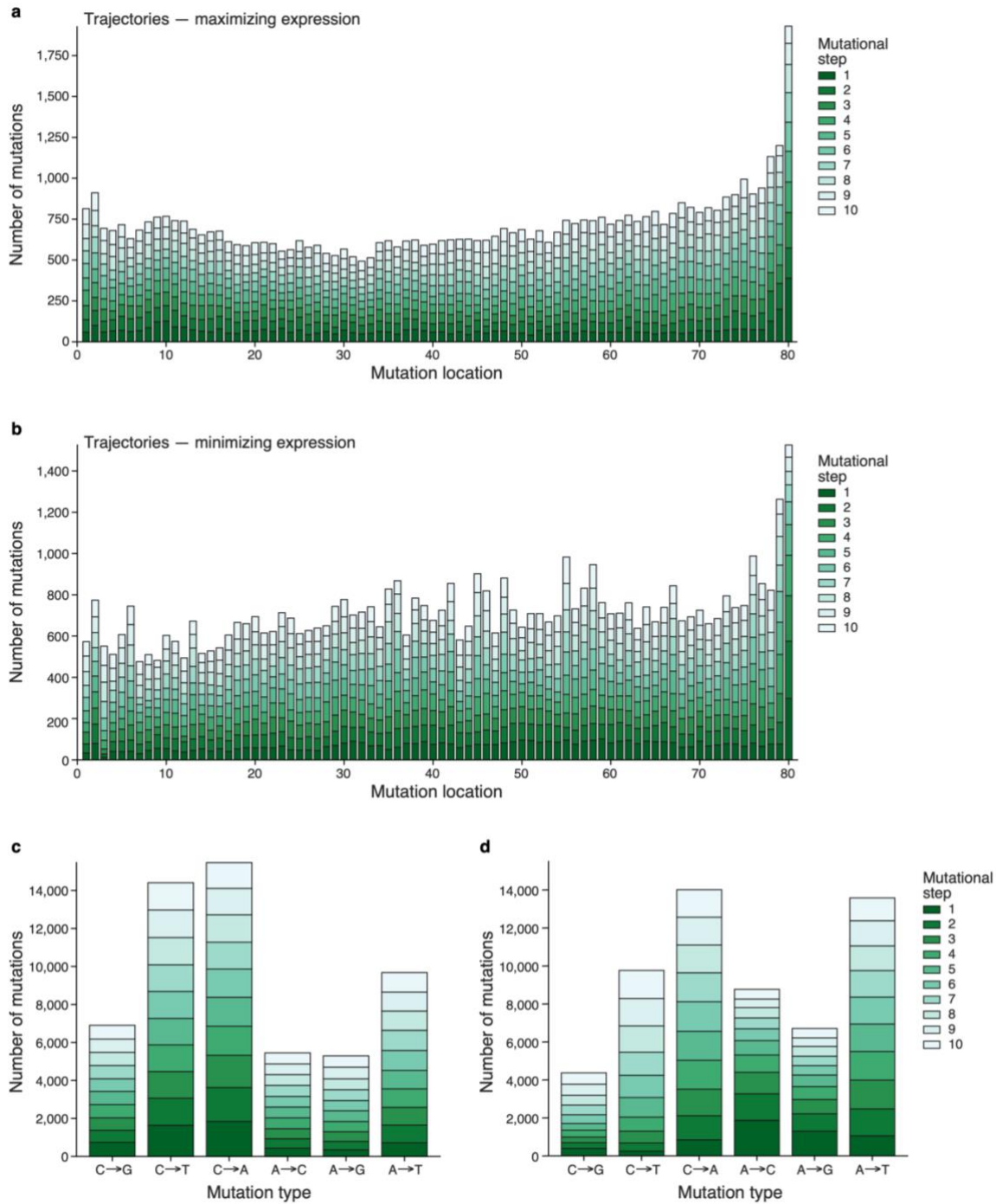
**Supplementary Fig. 15 | Comparison between predictions and measurements of expression at each mutational step in defined media.** Transformer model predicted ( $x$ -axes) and measured ( $y$ -axes) expression for each mutational step (plots) for trajectories in **Extended Data Fig. 1g** ( $n=6,304$  sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. Due to limitations in the number of sequences we could test per experiment, we only tested the decreasing expression defined media trajectory to 5 mutations. The two-tailed  $p$ -value corresponding to the performance metric shown in each panel is  $< 5 \cdot 10^{-234}$ .



**Supplementary Fig. 16 | Comparison between predictions and measurements of expression at each mutational step in defined media.** Convolutional model predicted ( $x$ -axes) and measured ( $y$ -axes) expression for each mutational step (plots) for trajectories in **Extended Data Fig. 1g** ( $n=6,304$  sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. Due to limitations in the number of sequences we could test per experiment, we only tested the decreasing expression defined media trajectory to 5 mutations. The two-tailed  $p$ -value corresponding to the performance metric shown in each panel is  $< 5 \cdot 10^{-234}$ .

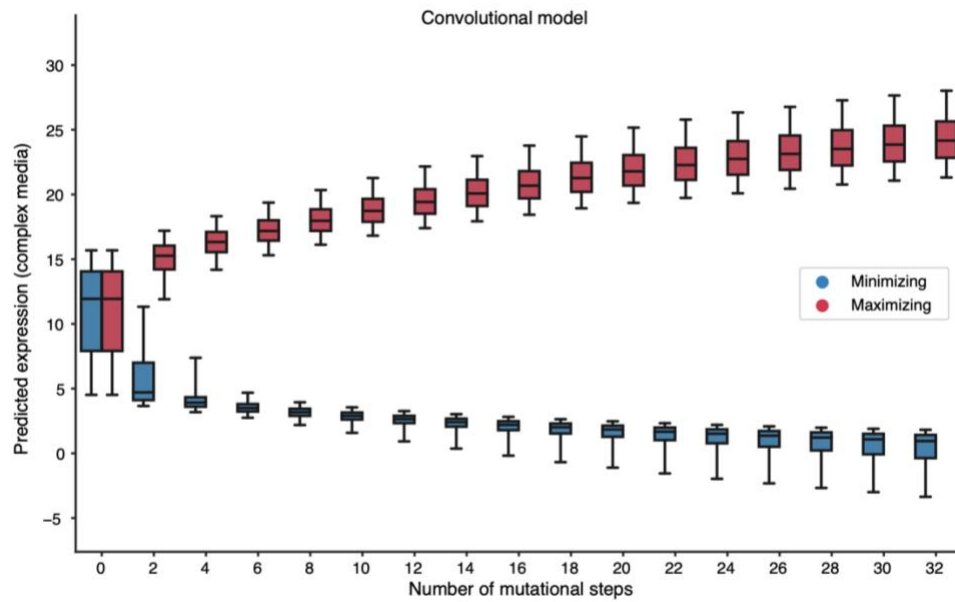


**Supplementary Data Fig. 17 | Characterization of sequence trajectories under strong competing selection pressures using the transformer model. a-d**, Competing expression objectives are slow to reach saturation. **a,b**, Difference in predicted expression (y axis) at each evolutionary time step (x axis) under selection to maximize (red) or minimize (blue) the difference between expression in defined and complex media, starting with either native sequences (**a**, n=5,720 trajectories) or random sequences (**b**, n=10,000 trajectories). **c-d**, Distribution of predicted expression (y axis) in complex (blue) and defined (red) media at each evolutionary time step (x axis) for a starting set of native sequences (**c**, n=5,720 trajectories) and random sequences (**d**, n=10,000 trajectories). Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **e** Motifs enriched within sequences evolved for competing objectives in different environments. Top five most enriched motifs, found using DREME(Bailey, 2011) (**Methods**) within sequences computationally evolved from a starting set of random sequences to either maximize (left) or minimize (right) the difference in expression between defined and complex media, along with DREME E-values, the corresponding rank of the same motif when using native sequences as a starting point, the likely cognate TF and that TF's known motif.

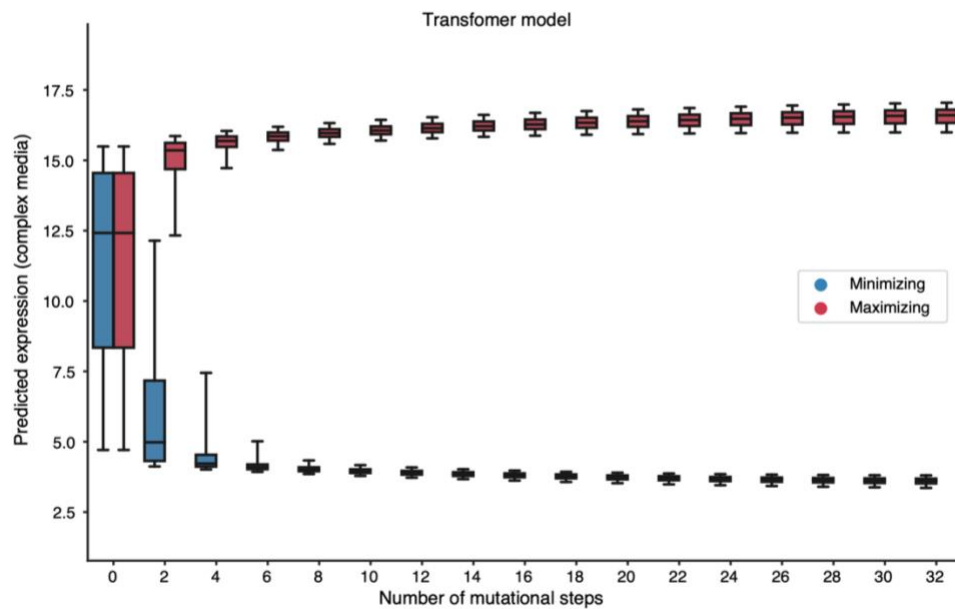


**Supplementary Fig. 18 | Sequences took diverse paths to evolve extreme expression in simulations with the transformer model. a-b,** The number (*y* axis) of mutations across each promoter position (*x* axis; -160 to -80 region) for trajectories under the SSWM regime starting with native promoter sequences when (a) maximizing or (b) minimizing expression in defined media using the transformer model. **c,d,** The number (*y* axis) of mutations of each type (*x* axis) for trajectories under the SSWM regime starting with native promoter sequences when (c) maximizing or (d) minimizing expression in defined media. Colors represent the mutational step (1-10).

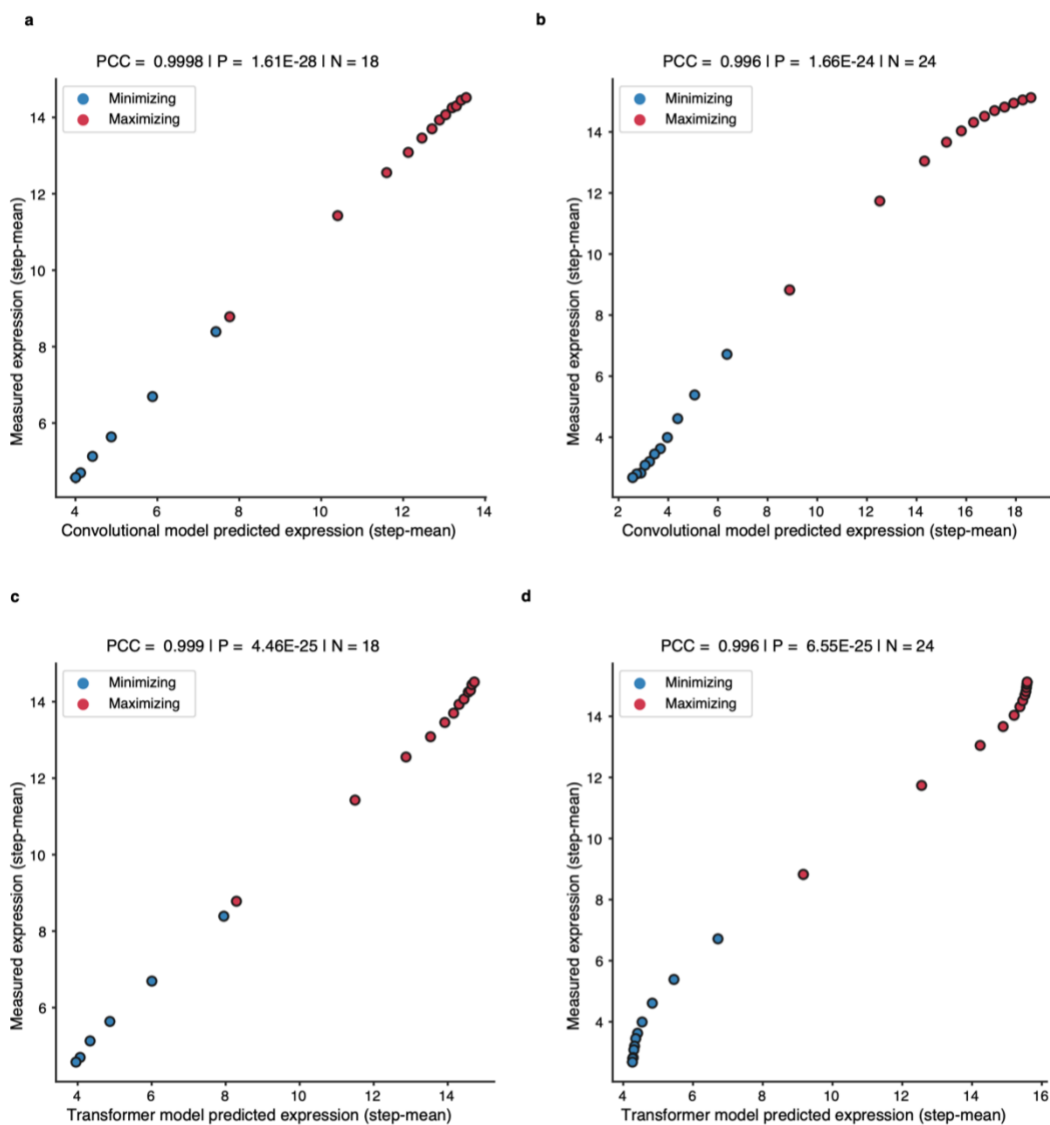
a



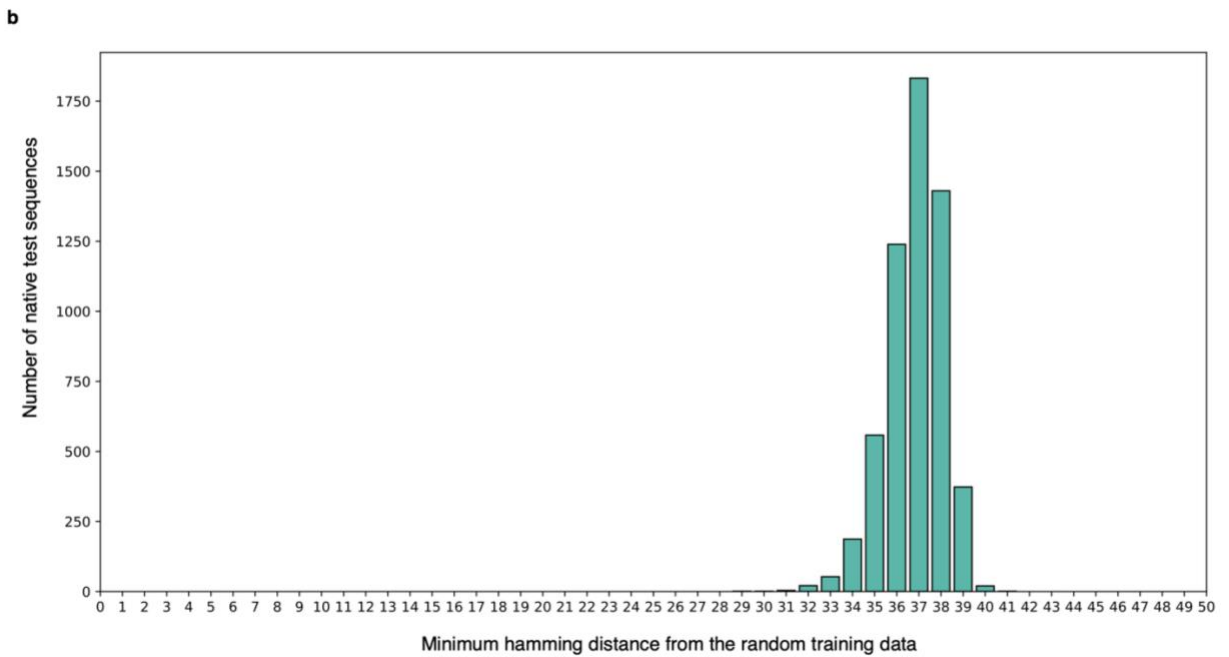
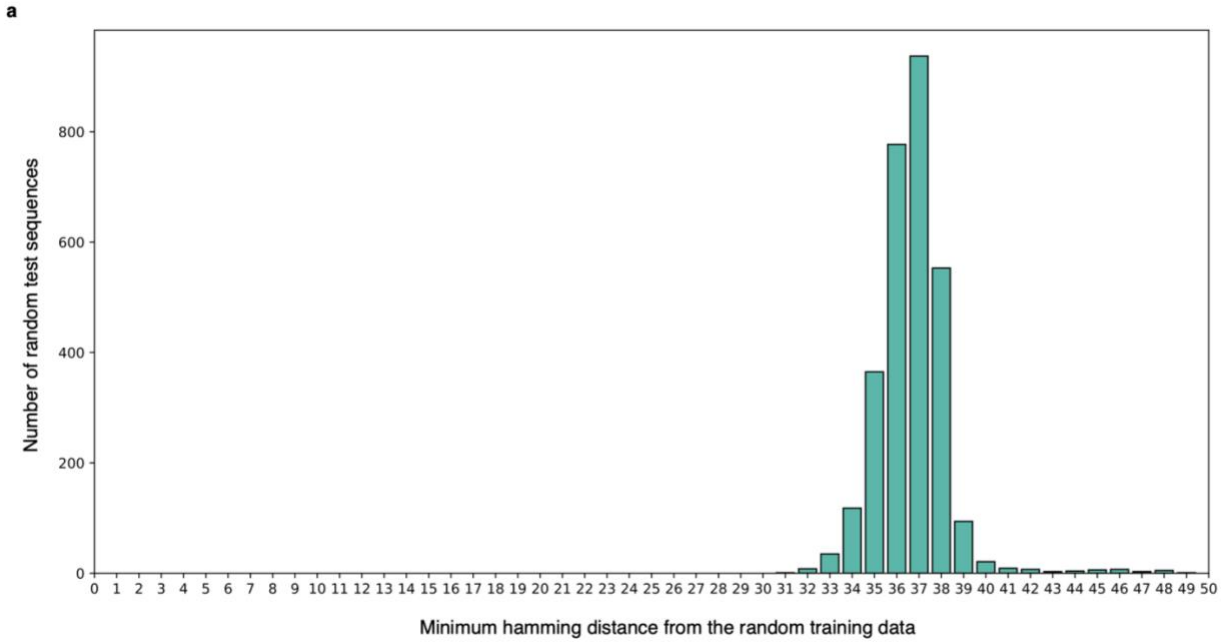
b



**Supplementary Fig. 19 | The transformer model captures expression plateau better than the convolutional model when simulating trajectories under SSWM for 32 mutational steps.** Distribution of predicted expression levels (y axis) in complex media at each mutational step (x axis) for sequence trajectories under SSWM favoring high (red) or low (blue) expression, starting with native promoter sequences using the convolutional (a, n=5,720 trajectories) or transformer (b, n=5,720 trajectories) models. The transformer model predicts an expression level plateau (like the measured expression in Fig. 2g), while the convolutional model predictions do not plateau at higher mutational distances. Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range.



**Supplementary Fig. 20 | Summary statistic scatterplot for trajectories under SSWM.** Mean measured expression ( $y$  axis) and mean predicted expression ( $x$  axis) at each step in the mutational trajectories for native sequences under SSWM for the convolutional (**a,b**) and transformer (**c,d**) model in the complex (**b,d**) and defined (**a,c**) media, as in Supplementary Fig. 19. The Pearson's correlation coefficient (PCC) and the corresponding two-tailed  $p$ -value are shown.



**Supplementary Fig. 21 | Sequence differences between training and test data.** The distribution of the Hamming distance between each sequence in the (a) random or (b) native test sets and the closest sequence in in the random training set.



## Supplementary Tables

### Supplementary Tables

**Supplementary Table 1** | The Expression Conservation Coefficient (ECC), mutational robustness, evolvability vector archetypal coordinates, predicted expression, ECC using gene-specific correction factors, and ECC non-neutrality p-values corresponding to all native promoter sequences.

**Supplementary Table 2** | The GO terms enriched by the ECC ranking. One-sided p-values were computed using minimum hypergeometric statistics, taking into account multiple testing as previously described (Eden *et al.*, 2009).

Supplementary Tables 1 and 2 are provided as an Excel file.

**Supplementary Table 3 | Primers used in this study.** The list of single stranded oligonucleotides used. This table can be found in the Supplementary Information document.

Name	Sequence (5'-3')	Orientation	Description	Reference
pCDC36_DBVPG6765_WT_fw	ATCCATACACAAGACTCATAGAA	Fw	WE gRNA	This study
pCDC36_DBVPG6765_WT_rv	AACTTCTATGAGTCTTGTGTATG	Rv	WE gRNA	This study
D6765_to_Y12_ssODN	TTCCATCTCTATATAACAAAGTAT TTCTTTATTTTCTAATAGTTCCTTT CTACGAGTCTTGTGTATGTTTATA AAGAGTGAGCTCTTTTGTATGAA GT	Duplex	ssODN SA allele	This study
pCDC36_seq_F	TCACACGTAGACGACTTGCCA	Fw	Sequencing	This study
pCDC36_seq_R2	CCTTGTAGTTTTTGCATATCTAGT	Rv	Sequencing	This study
Seq_3_Fw	ACTTGCCACATCCTGGTGTT	Fw	Sequencing	This study
Seq_3_Rv	ATGTTTCTGCCACGGTGAT	Rv	Sequencing	This study
CDC36_Fw	CATGACCTTAGGAGCGGACT	Fw	qPCR	This study
CDC36_Rv	TCCACTTCGCTTCTGGATGT	Rv	qPCR	This study
ACT1_Fw	TTGGCCGGTAGAGATTTGAC	Fw	qPCR	Teste et al.
ACT1_Rv	CCCAAAACAGAAGGATGGAA	Rv	qPCR	Teste et al.

RPN2_Fw	GCGGATACAGGCACATTGGATAC C	Fw	qPCR	Teste et al.
RPN2_Rv	TGTTGCTACCTTCTCTACCTCCTT ACC	Rv	qPCR	Teste et al.
pT-pA_GibsRI	GAACTGCATTTTTTTCACATCNNN NNNNNNNNNNNNNNNNNNNNNNNN NNNNNNNNNNNNNNNNNNNNNNNN NNNNNNNNNNNNNNNNNNNNNNNN NNNNNNNNNNNGGTTACGGCTGT TTCTTAA	Fw	Random promoter oligo for use in pTpA promoter context	de Boer et al
R-pT_GibsDS	TTAAGAAACAGCCGTAACC	Rv	For double- stranding pT- pA_GibsRI	de Boer et al
Nextera_i5LN5_GpT	TCGTCGGCAGCGTCAGATGTGTA TAAGAGACAGNNNNNTGCATTTT TTTCACATC	Fw	Nextera adaptor addition, with 5 random bases to help clustering	de Boer et al
Nextera_i7R_GpA	GTCTCGTGGGCTCGGAGATGTGT ATAAGAGACAGAACAGCCGTAAC C	Rv	Nextera adaptor addition	de Boer et al

**Supplementary Table 4 | Strains.** The list of yeast strains used.

Strain	Genotype	Reference
Y8205	<i>MATalpha, can1delta ::STE2pr-Sp_his5 lyp1delta ::STE3pr-LEU2 his3delta1 leu2delta0 ura3delta0</i>	Charles Boone Lab – strain verified by auxotrophy
S288C:: <i>ura3</i>	<i>MATα SUC2 gal2 mal2 mel flo1 flo8-1 hap1 ho bio1 bio6 ura3delta0</i>	de Boer et al. 2020 – strain verified by PCR of URA3
DBVPG6765 (WE)	<i>MATalpha, ho::NatMX, ura3::KanMX</i>	Cubillos, Louis & Liti (DOI: 10.1111/j.1567- 1364.2009.00583.x)
Y12 (SA)	<i>MATalpha, ho::NatMX, ura3::KanMX</i>	Cubillos, Louis & Liti
WE C7	DBVPG6765 derivate with SA Upc2 binding site	This study – pCDC36 genotype verified by Sanger sequencing
WE C23	DBVPG6765 derivate with SA Upc2 binding site	This study – pCDC36 genotype verified by Sanger sequencing

## References

- Agarwal V, Shendure J. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. 2020. *Cell Reports* 31 (7), 107663.
- Alipanahi, B. *et al.* (2015) “Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning,” *Nature Biotechnology*, 33(8), pp. 831–838.
- Bailey, T. L. (2011) “DREME: motif discovery in transcription factor ChIP-seq data,” *Bioinformatics (Oxford, England)*, 27(12), pp. 1653–1659.
- de Boer, C. G. *et al.* (2020) “Deciphering eukaryotic gene-regulatory logic with 100 million random promoters,” *Nature biotechnology*, 38(1), pp. 56–65.
- de Boer, C. G. and Hughes, T. R. (2012) “YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities,” *Nucleic Acids Res*, 40(Database issue), pp. D169–79.
- Brodsky, S. *et al.* (2020) “Intrinsically Disordered Regions Direct Transcription Factor In Vivo Binding Specificity,” *Molecular Cell*, 79(3), pp. 459–471.e4.
- Chen, J. *et al.* (2019) “A quantitative framework for characterizing the evolutionary history of mammalian gene expression,” *Genome research*, 29(1), pp. 53–63.
- De Boer, C. (2017) “High-efficiency *S. cerevisiae* lithium acetate transformation v1 (protocols.io.j4tcqwn),” *protocols.io*. ZappyLab, Inc. doi: 10.17504/protocols.io.j4tcqwn.
- Eden, E. *et al.* (2009) “GORilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists,” *BMC bioinformatics*, 10, p. 48.
- Keren, L. *et al.* (2016) “Massively Parallel Interrogation of the Effects of Gene Expression Levels on Fitness,” *Cell*, 166(5), pp. 1282–1294.e18.
- Langmead, B. *et al.* (2009) “Ultrafast and memory-efficient alignment of short DNA sequences to the human genome,” *Genome biology*, 10(3), p. R25.
- Li, J. *et al.* (2020) “DeepATT: a hybrid category attention neural network for identifying functional effects of DNA sequences,” *Briefings in bioinformatics*. doi: 10.1093/bib/bbaa159.
- Quang, D. and Xie, X. (2016) “DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences,” *Nucleic acids research*, 44(11), p. e107.
- Quang, D. and Xie, X. (2019) “FactorNet: A deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data,” *Methods (San Diego, Calif.)*, 166, pp. 40–47.
- Sharon, E. *et al.* (2012) “Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters,” *Nature biotechnology*, 30(6), pp. 521–530.
- Shrikumar, A., Greenside, P. and Kundaje, A. (2017) “Reverse-complement parameter sharing improves deep learning models for genomics,” *bioRxiv*, p. 103663.

Vaswani, A. *et al.* (2017) “Attention is All you Need,” in Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 5998–6008.

Weirauch, M. T. *et al.* (2013) “Evaluation of methods for modeling transcription factor sequence specificity,” *Nature Biotechnology*, 31(2), pp. 126–134.

Weirauch, M. T. and Hughes, T. R. (2010) “Conserved expression without conserved regulatory sequence: the more things change, the more they stay the same,” *Trends in genetics: TIG*, 26(2), pp. 66–74.

Zhou, J. and Troyanskaya, O. G. (2015) “Predicting effects of noncoding variants with deep learning-based sequence model,” *Nature methods*, 12(10), pp. 931–934.

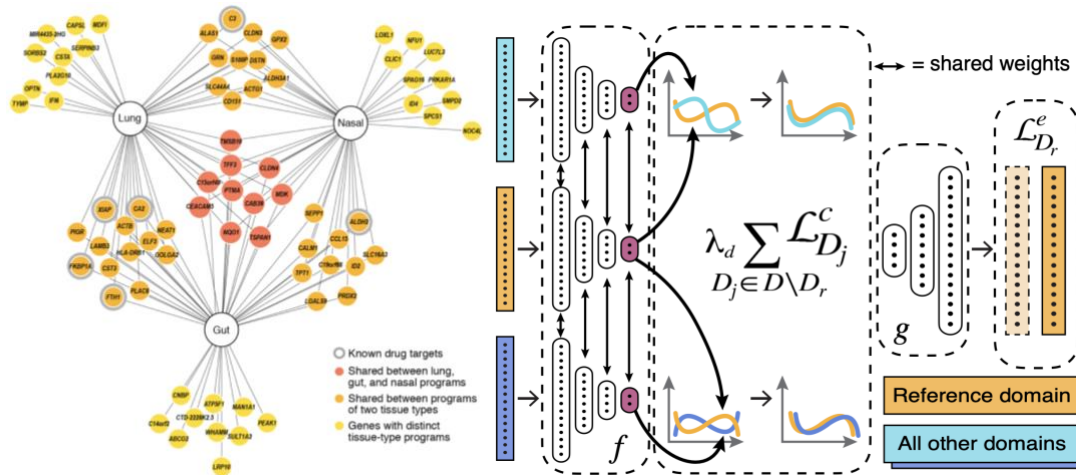


**Part B:**

*Expression*



## ATLAS (A Tool for Learning from Atlas-scale Single-cell measurements)



This chapter describes ‘A Tool for Learning from Atlas-scale Single-cell measurements’(ATLAS).

An early version of ATLAS appeared in:

*Single-cell meta-analysis of SARS-CoV-2 entry genes across tissues and demographics. **Nature Medicine** 27, 546–559 ([Muus, C. et al. 2021](#)). **Contribution:** Co-first author.*

Spatial mapping of cell types using ATLAS, was first presented in:

*Reference-based cell type matching of spatial transcriptomics data. **bioRxiv** 2022. ([Zhang et al](#)).*

**Contribution:** Co-author.

The manuscript that follows in this chapter, describes how ATLAS can be used to predict and prioritize drug targets from single cell atlases. **Contribution:** First author.

---

# ATLAS | How to predict and prioritize drug targets by expression program inference from single cell atlases

---

**Eeshit Dhaval Vaishnav**  
Massachusetts Institute of Technology  
edv@mit.edu

**Charles Comiter**  
Massachusetts Institute of Technology

**Ayshwarya Subramanian**  
Broad Institute of MIT and Harvard

**Karthik Jagadeesh**  
Broad Institute of MIT and Harvard

**Aviv Regev**  
Massachusetts Institute of Technology  
aviv.regev.sc@gmail.com

## Summary

The ongoing COVID-19 pandemic caused by the SARS-CoV-2 virus has sparked an urgent need for better understanding of its pathogenesis and identification of new drug targets to stem both viral infection and tissue-, organ- and body-level responses. Viral infection and host response begin at the single-cell level: the virus infects only specific cell types, while additional cell types respond to the intracellular signals triggered by infection. Integrative analyses of single-cell atlases can help identify both the specific cells involved and their therapeutically targetable programs. However, the complex and non-linear variability between measurements made across domains such as individuals, conditions, technological platforms and laboratories makes inference from these atlases challenging. Here, we introduce a novel framework for inference from atlas-scale scRNA-seq datasets. First, we learn biologically informative, domain-invariant feature representations for scRNAseq expression data by aligning domain distributions in a latent space through moment matching using a regularized autoencoder, correcting for undesirable variability across measurement domains. Then, we use these domain-invariant representations to identify gene programs with non-linear and combinatorial effects on phenotype using feature importance measures. We apply our framework to single cell atlases from autopsies and bronchoalveolar lavage samples from COVID-19 patients along with atlases from healthy individuals in the Human Cell Atlas to infer expression programs in SARS-CoV-2 target cells. We then predict and prioritize putative drug targets validated from independently published, drug repurposing and protein-protein interaction studies. This framework extends to both discrete (e.g. diseased vs healthy) and continuous (e.g. copy number variant scores) labels for scRNA-seq datasets and has broad applicability across a range of human diseases.

## 1 Contributions

We introduce a novel framework for predicting and prioritizing putative human disease drug targets from single-cell atlases that we call ATLAS (A Tool for Learning from Atlas-scale Single-cell datasets), with two primary contributions :

1. ATLASa : We propose a method for solving the scRNA-seq data-integration problem by using a higher order moment-matching regularizer with an AE to learn domain-invariant feature representations for scRNA-seq data.
  - We validate our approach on synthetic and real datasets and benchmark our performance against existing methods to demonstrate the quality of our learned representations.
  - We use these representations to identify and annotate cell types in a newly generated single-cell atlas from COVID-19 lung autopsies and a recently published lung atlas from healthy individuals [1].
2. ATLASb : We describe a simple approach for inferring gene expression programs from single-cell atlases using feature importance methods on models trained on our domain-invariant features that accounts for non-linear and combinatorial interactions between genes and their effects on phenotype.
  - We apply this approach to a recently published dataset of bronchoalveolar lavage fluid samples from COVID-19 patients [2] to infer cell-type specific gene expression programs
  - We identify drug targets, including ones that we were able to validate using independent experimental studies of SARS-CoV-2 protein-protein interaction (PPI) [3] and drug-repurposing [4]. We also demonstrate how to use our approach to prioritize putative drug target lists.

## 2 Approach

Our proposed method for learning domain-invariant features, ATLASa, is outlined in Figure 1a. We formulate this problem as a multi-target domain adaptation problem. The model architecture is that of an autoencoder comprised of an encoder  $f$  and a decoder  $g$  that are parametrized as a neural network with parameters  $\theta$ . The autoencoder has  $m$  streams with shared parameters as shown and each steam corresponds to an experimental domain.

**Input** : The model expects gene expression vectors for cells as the input. This input is supplied to the arm corresponding to the domain the cell is sampled from.

**Output** : The model outputs two domain invariant representations for each input  $x$ . The first output is  $f_\theta(x)$ , the latent feature representation learned by the encoder and, the second output is  $g_\theta(f_\theta(x))$ , a domain-invariant reconstruction of the input by the AE. We refer to this reconstruction in (ii) as the 'corrected' gene expression vector from here on out.

We minimize the pairwise domain discrepancy between an arbitrarily chosen reference domain and the rest by adapting the Central Moment Discrepancy (CMD) [5, 6] regularizer for matching the higher order central moments using order-wise moment differences for the latent feature space learned by  $f$ . The CMD regularizer is appropriate for this atlas-scale scRNA-seq data-integration problem because of its scalability (CMD computation is linear in the number of samples), minimal parameter sensitivity[5] and its ability to match non-Normal distributions.

### 2.1 Problem Formulation

Let  $D = \{D_j\}_{j=1}^m$  denote the set of scRNA-seq measurement domains.  $D_j = \{x_i^j\}_{i=1}^{n_j}$ , where each  $x_i^j$  refers to a single-cell gene expression vector measured in domain  $D_j$ . We first arbitrarily choose a reference domain  $D_r \in D$ . Then, for training, we sample mini-batches of size  $n_b$  from each domain  $X_j = \{x_i^j \in D_j \mid |X_j| = n_b\}$ . The objective function  $\mathcal{L}$  is a linear combination of the reference domain reconstruction loss and the pairwise CMD domain discrepancy losses for each non-reference domain w.r.t the reference domain.

$$\mathcal{L} = \mathcal{L}_{D_r}^e + \lambda_d \sum_{D_j \in D \setminus D_r} \mathcal{L}_{D_j}^c \quad (1)$$

$$\mathcal{L}_{D_r}^e(\theta) = \frac{1}{n_b} \sum_{x \in X_r} \|x - g_\theta(f_\theta(x))\|^2 \quad (2)$$

$$\mathcal{L}_{D_j}^c(\theta) = \frac{1}{|b-a|} \|E(f_\theta(X_j)) - E(f_\theta(X_r))\|_2 + \sum_{k=2}^K \frac{1}{|b-a|^k} \|C_k(f_\theta(X_j)) - C_k(f_\theta(X_r))\|_2 \quad (3)$$

where  $\|\cdot\|$  denotes the Euclidean norm,  $E(X) = \frac{1}{|X|} \sum_{x \in X} x$  is the empirical expectation vector, the parameter  $K$  is the bound on the order of the central moment terms,  $C_k(X) = E((x - E(X))^k)$  is the vector of all  $k^{th}$  order sample central moments,  $\mathcal{L}_{D_r}^c$  denotes the reference domain reconstructions loss and  $\mathcal{L}_{D_j}^c(\theta)$  denotes the domain discrepancy loss term for every other domain.

The objective function is minimized w.r.t.  $\theta$  using gradient based optimization methods.

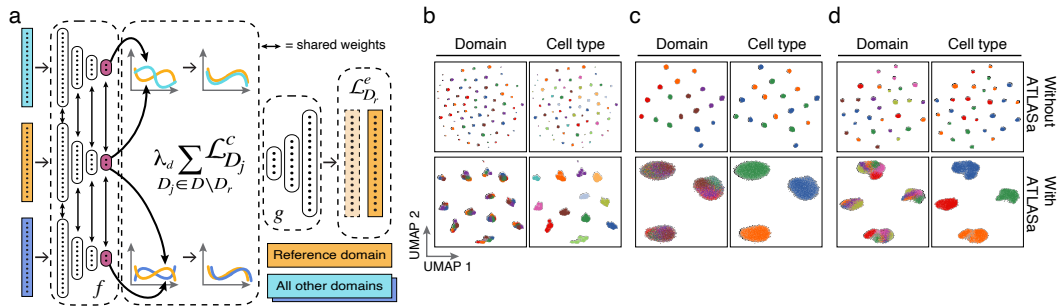
## 2.2 Datasets and Implementation Details

We use a letter code to refer to each dataset used in this manuscript. Complete details about accession (in the case of experimentally measured datasets) and simulations (in the case of synthetic datasets) will be made available with the Supplementary Materials. We used the following single-cell atlas datasets :

- COVIDA : A single-cell atlas of lung cells isolated from autopsy samples [citation TBD].
- COVIDB : A single-cell atlas of bronchoalveolar immune cells in COVID-19 patients [2].
- LUNG : A single-cell atlas of lung cells isolated from healthy individuals [1].
- SPLATTER2, SPLATTER4 and SPLATTER6 : Synthetic single-cell atlases that we generated for evaluation using a widely used scRNA-seq simulation library [7].
- SCIBSIM : A synthetic single-cell atlas used for benchmarking integration methods in [8].
- NASAL : A single-cell atlas generated from human surgical chronic rhinosinusitis tissue [9].

In addition to these single-cell atlases, we also used also used a dataset of SARS-CoV-2 protein-protein interactions (PPI) [3] and a drug-repurposing (REP) [4] study along with DRUGBANK [10], a comprehensive database of FDA approved and experimental drugs.

Details about the implementation, hyper-parameter considerations and dependencies can be found in the Supplementary Methods.



**Figure 1: ATLASa learns domain-invariant, biologically meaningful representations of single-cell data.** (a) A schematic outline of the ATLASa scRNA-seq data-integration method. Results of ATLASa applied to three simulated datasets (b, c, d) shown as 2-D UMAP visualizations. Each dot represents a cell, colored by domain of origin (left) or cell type (right). The top panels show the original simulated data UMAP visualizations without the ATLASa integration. The bottom panels show the same data with ATLASa integration. ATLASa integrated data shows clear separation of biologically relevant cell types when evaluated against the ground truth cell types used for the simulation, while displaying domain-invariance in the face of the original domain distribution discrepancy.

### 3 Results

#### 3.1 Performance Evaluation and Benchmarking

First, we qualitatively evaluate the performance of our scRNA-seq data-integration method ATLASa using three synthetic single-cell atlases (SPLATTER2, SPLATTER4, SPLATTER6). Each of these atlases were simulated to have pre-determined domain and cell type labels for each cell in the simulation to serve as ground truth in order to assess whether ATLASa learned domain-invariant representations  $f_{\theta}(x)$  and whether these still retained biological information. We ran ATLASa on each of these simulated atlases to learn domain-invariant representations for each cell and visualized them using a 2-dimensional UMAP [11] projection with and without using their learned representations using ATLASa. Figures 1b, c, d clearly show that the representations learned by ATLASa preserve biological information, as demonstrated by their ability to separate out biologically relevant cell-types in an unsupervised manner.

**Table 1:** Summary of performance on (domain-invariance, biological information preservation) metrics of an array of integration methods, including our ATLASa method. Entries take the form (KBET, ILASW). Both metrics are set up such that they range from 0 to 1 and higher values reflect better performance.

<i>Integration</i> <i>Dataset</i>	ATLASa	scVI	Scanorama	Harmony	None
NASAL	<b>(.632, .691)</b>	(.343, .616)	(.390, .559)	(.568, .673)	(.556, .676)
SCIBSIM	<b>(.999, .597)</b>	(.916, .573)	(.997, .546)	(.735, .537)	(.743, .551)
COVIDB	<b>(.283, .627)</b>	(.219, .557)	<b>(.379, .494)</b>	(.200, .553)	(.350, .568)
SPLATTER6	<b>(.969, .524)</b>	(.233, <b>.555</b> )	(.140, .518)	(.936, .527)	(.362, .514)
AVERAGE	<b>(.721, .610)</b>	(.428, .575)	(.476, .529)	(.466, .569)	(.646, .580)

Next, we quantitatively establish the efficacy of ATLASa, our proposed data-integration method, through a comprehensive benchmarking analysis against an array of established data-integration methods: single-cell Variational Inference (scVI) [?], Scanorama [12], and Harmony [13] (we refer the reader to the Supplement for further description of these methods and a comprehensive report of the benchmarking analysis). Since there may exist a trade-off between learning domain-invariant representations and preserving biologically meaningful information, our benchmarking analysis used two complementary metrics: (i) the k-Nearest-Neighbors Batch Effect Test (KBET) [14], a specially designed scRNA-seq data-integration assessment metric for evaluating the representations learned by the methods on their domain-invariance, and (ii) the Isolated Label Average Silhouette Width (ILASW) score [8], a metric designed for evaluating the preservation of biologically relevant information. We set up both metrics such that they lie in the range in from 0 to 1 and such that higher values on each are indicative of better performance and we refer the reader to the Supplementary Material for a complete description of these details.

**Table 2:** Summary of runtimes (in seconds) for ATLASa’s data integration method vs. scVI.

<i>Integration</i> <i>Dataset</i>	ATLASa	scVI
NASAL	<b>62.443</b>	1023.624
SCIBSIM	<b>70.817</b>	1105.963
COVIDB	<b>428.046</b>	2917.076
SPLATTER6	<b>110.230</b>	1808.429
AVERAGE	<b>167.884</b>	1713.773

Next, we benchmark the runtime of our method. At the time of benchmarking, out of all the methods considered, only scVI and ATLASa have GPU implementations. We compared their runtimes on each of the four datasets (see Table 2). Compared to scVI, ATLASa is over ten times faster on

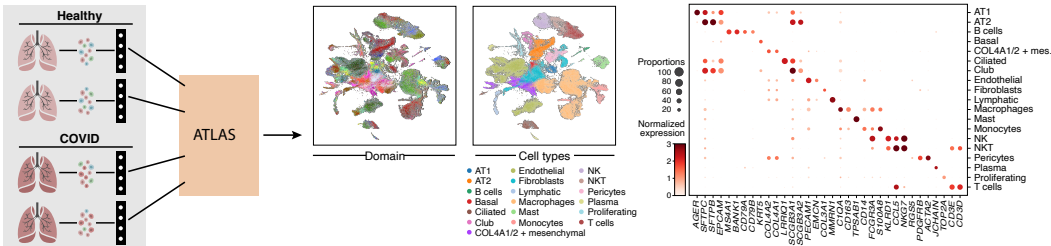
We ran our analysis on four datasets: two synthetic (SCIBSIM, SPLATTER6) and two biological (NASAL, COVIDB) (see **Datasets** and Supplementary Methods for further information). We found that on 3 out of 4 datasets, ATLASa’s data integration method outperformed the other methods on the domain-invariance metric (see **Table 1**). Furthermore, on 3 out of 4 datasets including the COVIDB dataset, ATLASa’s data integration method outperformed the others on the biological-information preservation metric. Importantly, ATLASa also had the highest average metric score on both of these complementary tasks (see **Table 1**).

average (in addition to being more effective on the domain-invariance and biological-information preservation tasks on all and all but one datasets, respectively, as shown in **Table 1**). One of the reasons for the runtime performance is the fact that CMD does not require computationally expensive kernel computations.

However, all of these metrics computed for various benchmarking tasks are just proxies for demonstrating potential utility of these approaches on real-life problems. Now, we turn our attention to a very real-life problem : COVID-19.

### 3.2 Identification and Annotation of Cell Types in a new COVID-19 Patient Single-Cell Atlas using Domain-Invariant representations from ATLASa

We use ATLASa to integrate two single cell atlases collected at opposite ends of the phenotype continuum: (i) A healthy human lung single cell atlas [11] spanning 17 experimental domains and 60,872 cells and (ii) A new lung cell atlas created from autopsies of COVID-19 patients ([citation pending]) representing 15 experimental domains and 27,519 cells. We then use the  $f_\theta(x)$  learned by ATLASa to identify and annotate cell types from these tissues to better understand COVID-19 biology using unsupervised clustering [Supplementary Materials]. We annotated the atlases using our results post-hoc using literature derived markers (Figure 2d) to define a shared taxonomy of 19 cell classes as shown in Figure 2c. The shared taxonomy we report includes Epithelial (AT1, AT2, Basal, Ciliated and Club cells), Stromal (Endothelial, Lymphatic, Mesenchymal) and Immune (Macrophages, Monocytes, Mast, T and B) cells. Further, Figure 2b demonstrates the domain-invariance of the learned representations from a diverse set of individuals displaying a broad range of phenotypes.

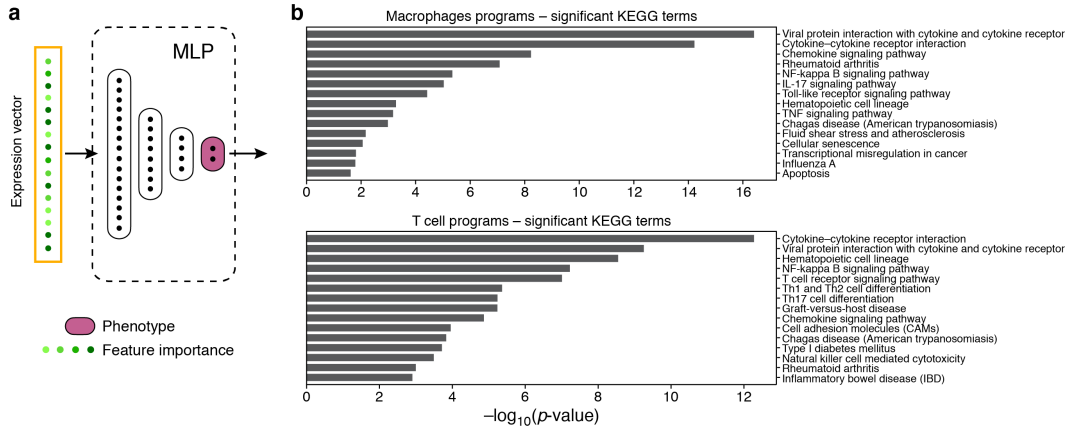


**Figure 2:** ATLASa identifies cell types in a new COVID-19 single-cell atlas. Demonstration of ATLASa for jointly analyzing two atlases. (a) Lung samples from (i) healthy and (ii) COVID-19 autopsy cohorts serve as input to ATLASa. Each lung sample represents an experimental domain. 2-D UMAP representations of the ATLAS integrated cells (dots) colored by domain of origin (b) and putative cell type (c). (d) Dotplot representation of marker genes used to annotate cell types. The size of the dot represents the proportion of cells expressing the marker gene (columns) and the color represents the average normalized gene expression in each cell type (rows). Normalized gene expression values are capped at 3.

### 3.3 Gene Expression Program Inference for COVID-19 from Healthy and Diseased Tissue Atlases

Genes can affect phenotypes in non-linear and combinatorial ways. Understanding these gene-expression programs (GEPs) that drive phenotypes of interest  $P$  can help elucidate the molecular mechanisms and pathways corresponding to disease states and facilitate drug target identification. Previous attempts at inferring GEPs for COVID19 relied heavily on distinguishing  $ACE2+TMPRSS2+$  cells (double positives, DPs) and compared these cells to  $ACE2-TMPRSS2-$  cells (double negatives, DNs) [15]. However these approaches were limited in their ability to identify GEPs relevant for drug target identification because of a lack of availability of single-cell atlases from COVID-19 patients. We use COVIDB, a recently published atlas of BALF samples from COVID-19 [2], to describe and demonstrate a simple approach (that we call ATLASb) for GEP inference from single-cell atlases.

ATLASb takes the ‘corrected’ gene expression vectors  $g_\theta(f_\theta(x))$  for each cell  $x$  produced by ATLASa as input to train a simple multi-layer perceptron (MLP) model  $F$  to predict phenotype  $P$  from  $g_\theta(f_\theta(x))$  such that  $F(g_\theta(f_\theta(x))) = P$ . Since the dimensions of the input of  $F$  correspond to a domain-invariant representations of gene expression vectors, we can now employ importance measures like SHAP values [16] and DeepLIFT [17] to identify gene expression programs driving



**Figure 3:** (a) Identifying GEPs using feature importance measures with MLPs. Gene programs inferred from the COVIDB dataset. KEGG gene set enrichment of *severe* (b) Macrophage and (c) T-cell expression programs. X-axis represents the enrichment test log p-values after adjustment for multiple hypothesis testing. Y-axis represents the enriched gene sets.

the phenotype of interest. This simple framework can allow us to identify genes that have non-linear and combinatorial effects on phenotypes because of the use of an MLP as the model  $F$ .

For the COVIDB dataset, we used a gradient-based approximation of SHAP values [16] with ATLASb to identify cell type specific GEPs over three different labeling regimes as phenotypes : *severity*, whether a cell is from a healthy patient or one with a severe case of COVID-19, *DP*, whether a cell has non-zero ACE2 and TMPRSS2 expression levels or zero for both, and *viral*, whether a cell has been infected by the virus or is simply a non-infected bystander [Supplementary Methods].

The GEPs allowed us to identify shared and cell type specific gene expression features ( Figure 3c, d). Overall, expression programs in all immune cells were characterized by enrichment for classic inflammatory molecules including *IL6*, members of the TNF family and complement component 3 (*C3*). The *severity* expression programs in macrophages were strongly enriched in cytokines and chemokines indicative of a pro-inflammatory “cytokine storm-like” state consistent with what has previously been reported [2]. On the other hand, the *viral* expression programs of macrophages included interferon regulatory genes *IRF8*, *IRF4* consistent with response to viral entry. Cell-type specific *severity* programs in epithelial cells include genes known to mediate viral infection (*CEACAM5*, *CLDN4*, *CLDN1*), and multi-functional cytokines including *IL10* and *CD40* signaling previously reported in inflamed airway cells [18]. Interestingly, *TMPRSS2* emerged as a highly discriminant gene, supporting its suggested role [19] as an accessory protease in mediating viral entry.

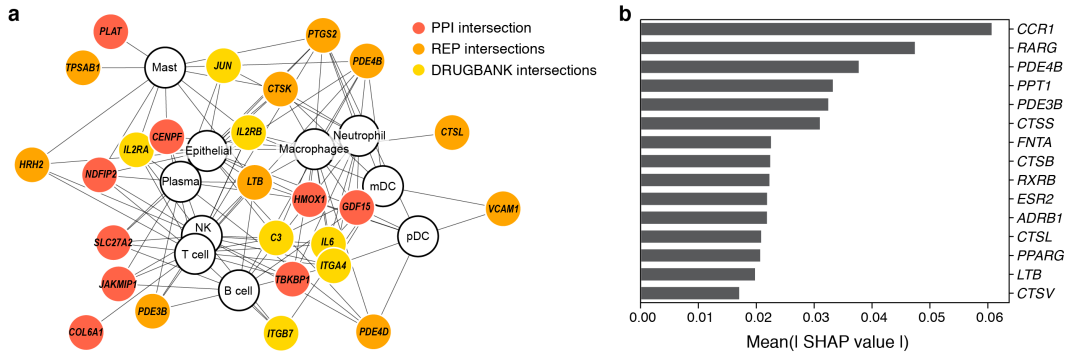
Taken together, this strong validation of our gene expression programs corroborated by multiple independent publications demonstrates the efficacy of our proposed ATLAS framework and its potential for driving biological discovery.

### 3.4 Predicting and Prioritizing Putative Drug Targets for COVID-19

Finally, since we had all the pieces in place to do so, we sought to evaluate whether our hypothesis that “domain-invariant feature representations of scRNA-seq data can help address the pressing need for identification of human disease drug targets when used in tandem with modern feature importance methods to account for the combinatorial and non-linear effects of gene-expression on phenotype” holds true in the context of COVID-19. We identified two independent validation datasets from a growing body of work on drug-screens and protein-protein interaction studies for COVID-19 : a dataset of SARS-CoV-2 protein-protein interactions (PPI) [3] and another from a SARS-CoV-2 drug-repurposing (REP) [4] study. We found multiple overlapping genes between our gene expression programs and the lists of drug targets identified in both of these independent experimental studies shown in Figure 4a. We also show in Figure 4a that our results hold across cell types suggesting that the putative drugs may potentially have mechanisms of action that could make them work across cell types. Figure 4a also shows a subset of its overlapping genes with Drugbank [10], a comprehensive

database of FDA approved and experimental drugs which may suggest novel putative targets with known drug-target interaction that were identified from scRNA-seq data using ATLAS and may have been missed by PPI studies potentially because of their indirect mechanism of action. Further inspection of Figure 4a shows that besides having the known contender IL6 which serves as further validation of our approach, the target lists also contain other putative hits including the complement component C3, and the inflammation-induced mediator of tissue tolerance GDF15 [20]. Other interesting drug targets include CDK6, which has not been previously reported in the context of SARS-CoV-2 but it has been reported as an interactor for viral cyclins [21].

These results suggest that our hypothesis may hold true.



**Figure 4:** (a) Network plot showing a subset of the drug targets discovered by ATLAS and their source of independent experimental validation : DRUGBANK [10] (a comprehensive list of FDA approved and experimental drug targets) intersections shown in yellow, COVID-19 REP (a large scale SARS-CoV-2 drug repurposing study) intersections in orange and the COVID-19 PPI (protein-protein interactions) intersections in red. (b) Prioritizing putative drug targets identified in the COVID-19 REP publication by ranking them using their feature SHAP importance values for predicting disease phenotypes.

For the development of effective therapeutics for COVID-19, putative drug targets must be prioritized in concert with the cellular context and function. Cell-type specific gene expression programs from COVID-19 data afford one window into both cellular localization and putative functional context for such prioritization. We show how one can prioritize the drug targets identified by the REP study (Figure 4b) by layering on the information from the COVIDB single-cell atlas using the same simple ATLASb framework described above. Among the top hits from this prioritization approach were Cathepsins *CTSS*, *CTSB*, *CTSL*, lysosomal genes essential for the cellular entry of coronaviruses [22], and previously proposed as targets for SARS-CoV [23]. Recently, amantadine was identified as an inhibitor of *CTSL*, and proposed [24] as a potential therapeutic for COVID-19 as well. Taken together, these independent sources from literature suggest that our approach for prioritizing putative COVID-19 drug targets may provide useful information for further follow-up with experimental studies.

## References

- [1] Kyle J Travaglini, Ahmad N Nabhan, Lolita Penland, Rahul Sinha, Astrid Gillich, Rene V Sit, Stephen Chang, Stephanie D Conley, Yasuo Mori, Jun Seita, Gerald J Berry, Joseph B Shrager, Ross J Metzger, Christin S Kuo, Norma Neff, Irving L Weissman, Stephen R Quake, and Mark A Krasnow. A molecular cell atlas of the human lung from single cell RNA sequencing. August 2019.
- [2] Mingfeng Liao, Yang Liu, Jing Yuan, Yanling Wen, Gang Xu, Juanjuan Zhao, Lin Cheng, Jinxiu Li, Xin Wang, Fuxiang Wang, Lei Liu, Ido Amit, Shuye Zhang, and Zheng Zhang. Single-cell landscape of bronchoalveolar immune cells in patients with covid-19. *Nature Medicine*, page 174, May 2020.
- [3] David E Gordon, Gwendolyn M Jang, Mehdi Bouhaddou, Jiewei Xu, Kirsten Obernier, Kris M White, Matthew J O'Meara, Veronica V Rezelj, Jeffrey Z Guo, Danielle L Swaney, Tia A Tummino, Ruth Huettnerhain, Robyn M Kaake, Alicia L Richards, Beril Tutuncuoglu, Helene



Foussard, Jyoti Batra, Kelsey Haas, Maya Modak, Minkyu Kim, Paige Haas, Benjamin J Polacco, Hannes Braberg, Jacqueline M Fabius, Manon Eckhardt, Margaret Soucheray, Melanie J Bennett, Merve Cakir, Michael J McGregor, Qiongyu Li, Bjoern Meyer, Ferdinand Roesch, Thomas Vallet, Alice Mac Kain, Lisa Miorin, Elena Moreno, Zun Zar Chi Naing, Yuan Zhou, Shiming Peng, Ying Shi, Ziyang Zhang, Wenqi Shen, Ilsa T Kirby, James E Melnyk, John S Chorba, Kevin Lou, Shizhong A Dai, Inigo Barrio-Hernandez, Danish Memon, Claudia Hernandez-Armenta, Jiankun Lyu, Christopher J P Mathy, Tina Perica, Kala B Pilla, Sai J Ganesan, Daniel J Saltzberg, Ramachandran Rakesh, Xi Liu, Sara B Rosenthal, Lorenzo Calviello, Srivats Venkataramanan, Jose Liboy-Lugo, Yizhu Lin, Xi-Ping Huang, Yongfeng Liu, Stephanie A Wankowicz, Markus Bohn, Maliheh Safari, Fatima S Ugur, Cassandra Koh, Nastaran Sadat Savar, Quang Dinh Tran, Djozhkun Shengjuler, Sabrina J Fletcher, Michael C O’Neal, Yiming Cai, Jason C J Chang, David J Broadhurst, Saker Klippsten, Phillip P Sharp, Nicole A Wenzell, Duygu Kuzuoglu, Hao-Yuan Wang, Raphael Trenker, Janet M Young, Devin A Cavero, Joseph Hiatt, Theodore L Roth, Ujjwal Rathore, Advait Subramanian, Julia Noack, Mathieu Hubert, Robert M Stroud, Alan D Frankel, Oren S Rosenberg, Kliment A Verba, David A Agard, Melanie Ott, Michael Emerman, Natalia Jura, Mark von Zastrow, Eric Verdin, Alan Ashworth, Olivier Schwartz, Christophe d’Enfert, Shaeri Mukherjee, Matt Jacobson, Harmit S Malik, Danica G Fujimori, Trey Ideker, Charles S Craik, Stephen N Floor, James S Fraser, John D Gross, Andrej Sali, Bryan L Roth, Davide Ruggero, Jack Taunton, Tanja Kortemme, Pedro Beltrao, Marco Vignuzzi, Adolfo Garcia-Sastre, Kevan M Shokat, Brian K Shoichet, and Nevan J Krogan. A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. *Nature*, April 2020.

- [4] Laura Riva, Shuofeng Yuan, Xin Yin, Laura Martin-Sancho, Naoko Matsunaga, Sebastian Burgstaller-Muehlbacher, Lars Pache, Paul P De Jesus, Mitchell V Hull, Max Chang, Jasper Fuk-Woo Chan, Jianli Cao, Vincent Kwok-Man Poon, Kristina Herbert, Tu-Trinh Nguyen, Yuan Pu, Courtney Nguyen, Andrey Rubanov, Luis Martinez-Sobrido, Wen-Chun Liu, Lisa Miorin, Kris M White, Jeffrey R Johnson, Christopher Benner, Ren Sun, Peter G Schultz, Andrew Su, Adolfo Garcia-Sastre, Arnab K Chatterjee, Kwok-Yung Yuen, and Sumit K Chanda. A large-scale drug repositioning survey for SARS-CoV-2 antivirals. April 2020.
- [5] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for Domain-Invariant representation learning. February 2017.
- [6] Werner Zellinger, Bernhard A Moser, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Robust unsupervised domain adaptation for neural networks via moment alignment. *Inf. Sci.*, 483:174–191, May 2019.
- [7] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: Simulation of single-cell rna sequencing data. *Genome Biology*, page 174, September 2017.
- [8] M D Luecken, M Büttner, K Chaichoompu, A Danese, M Interlandi, M F Mueller, D C Strobl, L Zappia, M Dugas, M Colomé-Tatché, and F J Theis. Benchmarking atlas-level data integration in single-cell genomics. May 2020.
- [9] Jose Ordovas-Montanes, Daniel F Dwyer, Sarah K Nyquist, Kathleen M Buchheit, Marko Vukovic, Chaarushena Deb, Marc H Wadsworth, 2nd, Travis K Hughes, Samuel W Kazer, Eri Yoshimoto, Katherine N Cahill, Neil Bhattacharyya, Howard R Katz, Bonnie Berger, Tanya M Laidlaw, Joshua A Boyce, Nora A Barrett, and Alex K Shalek. Allergic inflammatory memory in human respiratory epithelial progenitor cells. *Nature*, 560(7720):649–654, August 2018.
- [10] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.*, 46(D1):D1074–D1082, January 2018.
- [11] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. February 2018.
- [12] Bonnie Berger Brian Hie, Bryan Bryson. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Naure Biotechnology*, 37(June):685–691, 2019.

- [13] Ilya Korsunsky, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-Ru Loh, and Soumya Raychaudhuri. Fast, sensitive, and accurate integration of single cell data with harmony.
- [14] Maren Büttner, Zhichao Miao, F Alexander Wolf, Sarah A Teichmann, and Fabian J Theis. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods*, 16(1):43–49, January 2019.
- [15] Christoph Muus, Malte D Luecken, Gokcen Eraslan, Avinash Waghay, Graham Heimberg, Lisa Sikkema, Yoshihiko Kobayashi, Eeshit Dhaval Vaishnav, Ayshwarya Subramanian, Christopher Smilie, Karthik Jagadeesh, Elizabeth Thu Duong, Evgenij Fiskin, Elena Torlai Triglia, Meshal Ansari, Peiwen Cai, Brian Lin, Justin Buchanan, Sijia Chen, Jian Shu, Adam L Haber, Hattie Chung, Daniel T Montoro, Taylor Adams, Hananeh Aliee, J Samuel, Allon Zaneta Andrusivova, Ilias Angelidis, Orr Ashenberg, Kevin Bassler, Christophe Bécavin, Inbal Benhar, Joseph Bergensträhle, Ludvig Bergensträhle, Liam Bolt, Emelie Braun, Linh T Bui, Mark Chaffin, Evgeny Chichelnitskiy, Joshua Chiou, Thomas M Conlon, Michael S Cuoco, Marie Deprez, David S Fischer, Astrid Gillich, Joshua Gould, Minzhe Guo, Austin J Gutierrez, Arun C Habermann, Tyler Harvey, Peng He, Xiaomeng Hou, Lijuan Hu, Alok Jaiswal, Peiyong Jiang, Theodoros Kappellos, Christin S Kuo, Ludvig Larsson, Michael A Leney-Greene, Kyungtae Lim, Monika Litviňuková, Ji Lu, Leif S Ludwig, Wendy Luo, Henrike Maatz, Elo Madisson, Lira Mamanova, Kasidet Manakongtreecheep, Charles-Hugo Marquette, Ian Mbano, Alexi Marie McAdams, Ross J Metzger, Ahmad N Nabhan, Sarah K Nyquist, Lolita Penland, Olivier B Poirion, Sergio Poli, Cancan Qi, Rachel Queen, Daniel Reichart, Ivan Rosas, Jonas Schupp, Rahul Sinha, Rene V Sit, Kamil Slowikowski, Michal Slyper, Neal Smith, Alex Sountoulidis, Maximilian Strunz, Dawei Sun, Carlos Talavera-López, Peng Tan, Jessica Tantivit, Kyle J Travaglini, Nathan R Tucker, Katherine Vernon, Marc H Wadsworth, Julia Waldman, Xiuting Wang, Wenjun Yan, William Zhao, Carly G K Ziegler, The NHLBI LungMAP Consortium, and The Human Cell Atlas Lung Biological Network. Integrated analyses of single-cell atlases reveal age, gender, and smoking status associations with cell type-specific expression of mediators of SARS-CoV-2 viral entry and highlights inflammatory programs in putative target cells. April 2020.
- [16] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.
- [17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. April 2017.
- [18] Francesca Cagnoni, Susanna Oddera, Julien Giron-Michel, Anna Maria Riccio, Susanna Olsson, Palmiro Dellacasa, Giovanni Melioli, G. Walter Canonica, and Bruno Azzarone. Cd40 on adult human airway epithelial cells: Expression and proinflammatory effects. *The Journal of Immunology*, 172(5):3205–3214, 2004.
- [19] Markus Hoffmann, Hannah Kleine-Weber, Simon Schroeder, Nadine Krüger, Tanja Herrler, Sandra Erichsen, Tobias S. Schiergens, Georg Herrler, Nai-Huei Wu, Andreas Nitsche, Marcel A. Müller, Christian Drosten, and Stefan Pöhlmann. Sars-cov-2 cell entry depends on ace2 and tmprss2 and is blocked by a clinically proven protease inhibitor. *Cell*, 181(2):271 – 280.e8, 2020.
- [20] Harding H. Luan, Andrew Wang, Brandon K. Hilliard, Fernando Carvalho, Connor E. Rosen, Amy M. Ahasic, Erica L. Herzog, Insoo Kang, Margaret A. Pisani, Shuang Yu, Cuiling Zhang, Aaron M. Ring, Lawrence H. Young, and Ruslan Medzhitov. Gdf15 is an inflammation-induced central mediator of tissue tolerance. *Cell*, 178(5):1231 – 1244.e11, 2019.
- [21] Ursula Schulze-Gahmen and Sung-Hou Kim. Structural basis for CDK6 activation by a virus-encoded cyclin. *Nat. Struct. Biol.*, 9(3):177–181, March 2002.
- [22] Christine Burkard, Monique H Verheije, Oliver Wicht, Sander I van Kasteren, Frank J van Kuppeveld, Bart L Haagmans, Lucas Pelkmans, Peter J M Rottier, Berend Jan Bosch, and Cornelis A M de Haan. Coronavirus cell entry occurs through the endo-/lysosomal pathway in a proteolysis-dependent manner. *PLoS Pathog.*, 10(11):e1004502, November 2014.
- [23] Graham Simmons, Dhaval N. Gosalia, Andrew J. Rennekamp, Jacqueline D. Reeves, Scott L. Diamond, and Paul Bates. Inhibitors of cathepsin l prevent severe acute respiratory syndrome

coronavirus entry. *Proceedings of the National Academy of Sciences*, 102(33):11876–11881, 2005.

- [24] Sandra P Smieszek, Bart P Przychodzen, and Mihael H Polymeropoulos. “amantadine disrupts lysosomal gene expression; potential therapy for COVID19”. April 2020.

---

# Supplementary Information

## ATLAS | How to predict and prioritize drug targets by expression program inference from single cell atlases

---

### Contents

<b>1 Introduction to the Supplementary Materials</b>	<b>1</b>
<b>2 Definitions</b>	<b>2</b>
<b>3 Datasets</b>	<b>2</b>
<b>4 Implementation, Preprocessing and Hyperparameters</b>	<b>3</b>
<b>5 Performance Evaluation and Benchmarking</b>	<b>6</b>
<b>6 Gene Expression Program Inference for COVID-19 from Healthy and Diseased Tissue Atlases</b>	<b>7</b>
<b>7 Errata and Clarifications</b>	<b>10</b>

## 1 Introduction to the Supplementary Materials

In this document, we elaborate on the details omitted from the main text of the manuscript. Additionally, we will provide further clarification on the methods and results. Source code for ATLAS and all the analyses reported in this manuscript can be found in the files attached in the zipped folder (with the specific references to the files below). The datasets used are made available on a [Google Bucket](#) (with the exception of COVIDA, which is currently not publicly available). We begin by briefly summarizing our primary results :

In the main text, we introduced ATLAS: A Tool for Learning from Atlas-scale Single-cell datasets. ATLAS is a two part framework consisting of ATLASa and ATLASb. We used ATLASa to learn domain-invariant representations of scRNA-seq datasets and demonstrated the method's effectiveness qualitatively (in the main text's Figure 1) and quantitatively (in the main text's Table 1 and Table 2) using real and simulated datasets. Then, we demonstrated ATLASa's applicability to COVID-19 by using it to identify cell types from single-cell atlases constructed from autopsies of COVID-19 patients by integrating them with atlases from healthy control lung samples from [\[1\]](#). Next, we developed ATLASb, an approach for inferring gene-expression programs driving a phenotype (like disease state) using scRNA-seq atlases. Using ATLASb in tandem with ATLASa on a recently published COVID-19 single-cell atlas, we identified gene expression programs and drug targets, both of which we validated using multiple independent published studies.

## 2 Definitions

In this section, we define the terms we used throughout the main text and the supplement :

- **COVID-19** : Coronavirus Disease 2019
- **SARS-CoV-2** : Severe Acute Respiratory Syndrome Coronavirus 2
- **Gene** : A sequence of DNA characters called nucleotides that codes for the synthesis of gene products like proteins.
- **Protein** : A gene product that is a polymer made from amino acids that often has a defined structure and function.
- **Phenotype** : A set of observable traits of an organism that is a function of an organism’s gene expression profile and its environment.
- **Single-Cell Sequencing** : A set of technologies developed for making measurements from individual cells.
- **Single-cell RNA Sequencing (scRNA-seq)** : A technology that quantifies the gene expression from individual cells by measuring the amount of messenger RNA produced by all the genes in the cell.
- **Drug Targets** : A molecule in the body (often a protein, which is the product of the expression of a gene) that is strongly associated with a disease phenotype and that may be perturbed by a drug to produce a therapeutic effect.
- **Gene Expression Program (GEP)** : In this context, the set of genes that drive the corresponding phenotype of interest.
- **KEGG term** : KEGG (Kyoto Encyclopedia of Genes and Genomes) [2] is a widely used database for interpreting genomic data. KEGG terms refer to the molecular functions of individual genes and sets of genes (e.g. : a GEP).
- **FDA** : The Food and Drug Administration, a United States federal executive department responsible for the process of approving drugs.

## 3 Datasets

Here, we elaborate on the dataset descriptions provided in the main text. All simulations and published experimental datasets we used are made available as `.h5ad AnnData` objects [3] that can be found at [https://console.cloud.google.com/storage/browser/atlas\\_datasets/](https://console.cloud.google.com/storage/browser/atlas_datasets/). The name of each available dataset in the list below may be clicked-on for a direct download link to a pre-processed version of each dataset with the non-preprocessed, raw counts in the `AnnData.layers['counts']` field :

- [SPLATTER6](#), [SPLATTER4](#), [SPLATTER2](#) | Synthetic single-cell atlases that we generated for evaluation using a widely used scRNA-seq simulation tool `SpLatteR` [4]. [SPLATTER6](#) (shown in Figure 1b in the main text), [SPLATTER4](#) (shown in Figure 1c in the main text), and [SPLATTER2](#) (shown in Figure 1d in the main text) consisted of 23148, 19318, and 15441 cells, respectively, each with 9987, 9983, and 9974 genes, respectively. The datasets were simulated to have come from six, six and eight experimental domains, respectively and to have twelve, three, and four celltypes, respectively. The `AnnData.obs.Batch` field contains the domain label for each cell (in each of these three datasets) and similarly the `AnnData.obs.Group` field contains the cell type label for each cell.
- [NASAL](#) | A single-cell atlas generated from human surgical chronic rhinosinusitis tissue [5]. It consists of 7087 cells, each with 33694 measured genes. The `AnnData.obs.donor` field contains the domain label for each cell and the `AnnData.obs.ann_level_4` field contains the cell type label for each cell.
- [SCIBSIM](#) | A synthetic single-cell atlas used for benchmarking integration methods based on simulations from [6] (their manuscript contains further details on how they simulated this dataset).

It contains 20100 cells, each with 10000 genes. The `AnnData.obs.Batch` field contains the domain label for each cell and the `obs.Group` field contains the cell type labels for each cell.

- **COVIDB** | A recently published single-cell atlas of bronchoalveolar immune cells in COVID-19 patients [7]. It consists of 63103 cells, each with 33538 genes. The `AnnData.obs.sample_new` field contains the domain label for each cell and the `AnnData.obs.celltype` field denotes the cell type (out of the ten celltypes reported in the publication) that each cell belongs to.
- Datasets used in Figure 3 : We used two single-cell atlases for Figure 3. The first (COVIDA) is from an ongoing effort to generate atlases from lung autopsies of COVID-19 patients and the second (LUNG) is an atlas generated from lung samples isolated from healthy individuals.
  - COVIDA - A single-cell atlas of lung cells isolated from autopsy samples [citation TBD]. It consists of 34523 cells, each with 28560 genes. We will make a data download link available for this dataset when the full dataset and the associated citation becomes publicly available. We will also update this section of the supplement with a final citation (currently pending) when it is possible to appropriately attribute everyone who generated this growing single-cell atlas.
  - LUNG - A single-cell atlas of lung cells isolated from healthy individuals [1]. It consists of 60993 cells, each with 26485 genes. This dataset is available at <https://hlca.ds.czbiohub.org/>

## 4 Implementation, Preprocessing and Hyperparameters

### Implementation

All code used is made available along with the Supplementary Materials in the ATLAS-master folder. The `README.md` file in this folder describes how to create an environment for using ATLAS with all the dependencies installed. A description of each file follows :

- `tables_12main.ipynb` : Notebook to produce corrected `AnnData` objects and performance metrics for ATLASa and a suite of other data-integration methods (from Table 1 and Table 2 of the main manuscript).
- `program_analysis.ipynb` : Notebook to produce cell-type specific gene programs for a variety of feature-importance and phenotype labeling regimes used in the analysis including Tables 1, 2 and 3 shown in this Supplement.
- `tables_123456supp_figures_4supp.ipynb` : Notebook to produce Tables 4, 5 and 6 in this Supplement and to find the intersections for the Venn Diagram in this Supplement's Figure 4.
- `figures_12main_123supp.ipynb` : Notebook to reproduce hyperparameter experiments, pre vs post ATLASa-correction UMAPs, and dot-plot associated with Figures 1 and 2 in the main text and Figures 1, 2, and 3 in this Supplement.
- `figures_34main.ipynb` : Notebook to produce the GEP bar-plots and drug-target network plot from Figures 3 and 4 in the main text of the manuscript.
- `correct_aux.py` : Backend of the ATLASa data-integration method and associated helper functions.
- `analysis_aux.py` : Backend of the ATLASb feature-importance drug-target-prediction method and associated helper functions.
- `atlas.py` : Programmer friendly API for ATLAS use on other scrRNA-seq atlases and disease labels (currently under development, to be officially released soon).

The ATLASa and ATLASb model implementations used `Tensorflow` [8] and `Keras` [9]. The training and evaluation were carried out on a NVIDIA Tesla M60 GPU. A machine with consistent hardware configurations was used for all the benchmarking experiments. All single-cell RNA-sequencing datasets were converted to `.h5ad AnnData` objects [3] before preprocessing as described below. The primary language of all the implementation was Python, with the exception of the simulated scrRNA-seq datasets that used R to interface with Splatter [4]. The network plot in Figure 4 was generated using

`networkx` and the KEGG term enrichment was performed using the `scanpy.queries.enrich` function in Scanpy [10].

## Preprocessing

For use as input with ATLASa, we started with raw count matrices from each atlas dataset above whenever those were available. Then, we normalized the counts such that the sum of expression values across genes for each cell was  $10^4$ . Then we log scaled the data after adding 1 to each entry in the gene expression matrix. When raw counts weren't available, we directly used the dataset with the transcripts per million (TPM) units made available with the published dataset. After this step, for all the datasets, we then normalized again to a target sum of 1, zero-centered and scaled the data to have mean 0 and unit variance, and clipped off the scaled values to restrict them to a maximum value of 10. We used the same preprocessing steps for all the external data-integration methods we benchmarked our method against with the exception of the methods that required raw, un-processed counts to be presented as input (scVI, in particular). The domain from which the largest number of cells originated was arbitrarily chosen as the Reference Domain  $D_r$  for training the model. When working with large-scale datasets, ATLASa can use the projection of the gene expression data along its first hundred orthonormal principal component vectors before the first layer of the AE and then use the corresponding inverse transform at the end of the output layer of the AE for speed and scalability. This option was used whenever the `hp_dict['use_rep']` value was set to 'X\_pca' in the code.

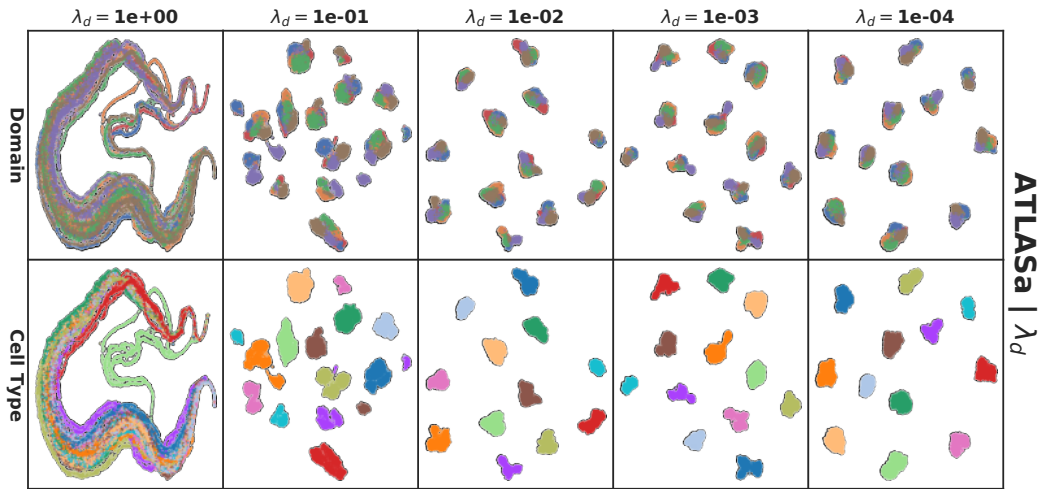
For use as input to ATLASb, we directly used the corrected gene expression vectors output by ATLASa for each cell as described in the main text. We note that the simple approach suggested in ATLASb is independent of preprocessing. Any form of the gene expression data may be used to train a classifier from gene expression matrices to predict phenotypes and this classifier may then be examined using variable importance measures to identify GEPs and drug targets as described in the main text. Here, the dimensions of the input correspond to the genes measured in the cell. To speed up ATLASb, one can use a subset of the genes that are highly variable (defined using the `scanpy.pp.highly_variable_genes` function) and/or the subset on the genes that are upregulated in expression (defined as the genes that have a higher mean expression in one phenotype class than another) in addition to differentially expressed genes (identified using the `scanpy.tl.rank_genes_groups` function). Finally, for the COVIDB dataset, the publication reported some ambient contamination (more details can be found in their [Data exclusions](#) section of their original publication [7]) and so we addressed that by adapting a procedure previously used in [11] for filtering out putative ambient RNA contaminants. This procedure gave us a list of contaminant genes for each cell type that we subsequently excluded from all further analysis using ATLASb on the COVIDB dataset.

## Hyperparameters

The model architecture consists of an autoencoder (AE) with multiple streams, each corresponding to a domain as described in Section 3 of the main text. The encoder consists of three fully-connected layers with [1024, 512, 258] neurons respectively. The decoder consists of two fully connected layers, the first of which has 512 neurons and the second one has the same dimensions as the input. Each fully connected layer (except the last one) is followed by a Parametric ReLu Activation layer [12] and each connection has a 0.1 probability of dropout. The reconstruction loss term on the cells from the reference domain is computed using the `mean_squared_error` function. For every other domain, the domain discrepancy term w.r.t. the reference domain is computed using the `cmd` function adapted from [13]. The objective function is constructed as described in the main text and is optimized using the Adam Optimizer [14]. The model was trained for 10 epochs with a mini-batch size of 32. The code corresponding to this section can be found in the `get_corrected_adata` function from the `correct_aux.py` file shared with the Supplementary Materials. The `hp_dict` variable for each dataset described in the code specifies the exact hyperparameters used for each experiment.

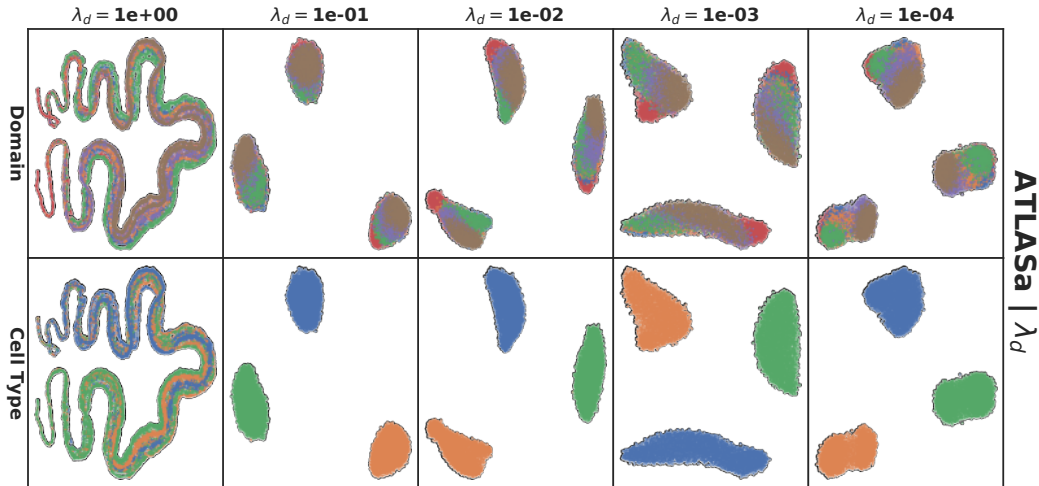
The hyperparameter  $\lambda_d$ , the weight of the sum of the distribution moment matching terms for each domain, was chosen after a series of experiments on the SPLATTER datasets. We started with a low value for  $\lambda_d$  and increased the value by an order of magnitude until it broke the model at  $\lambda_d = 1$  (evident from the lack of separation between the ground truth cell types in the simulation for  $\lambda_d = 1$  in Figures 1, 2 and 3 in this supplement). The range considered was [1e-4, 1e-3, 1e-2, 1e-1, 1

1



**Figure 1:** A range of  $\lambda_d$  values for SPLATTER6 with ATLASa

1



**Figure 2:** A range of  $\lambda_d$  values for SPLATTER4 with ATLASa

and all values of  $\lambda_d < 1$  performed very well on the domain-invariant representation learning task as shown in this Supplement's Figure 1, Figure 2 and Figure 3 by the separation of cell types across domains. The first row in each of these figures shows the cells colored by their experimental domain of origin and the second row shows cells colored by their biological cell type pre-defined in the simulation. Each column corresponds to a unique value of  $\lambda_d$ . We picked the middle of this range  $\lambda_d = 1e-2$  for all the subsequently analyzed COVID-19 relevant datasets whose results are presented in the manuscript. The uncorrected version of the data can be found in the top row of the main text Figures 1b, c and d. The bottom row of the main text Figures 1b, c and d were generated from scratch after the hyperparameter exploration using  $\lambda_d = 1e-2$ .

Finally, the hyperparameter  $K$ , the bound on the order of the moment central moment terms, is chosen as  $K = 3$  based on the hyperparameter sensitivity experiments in [13] showing that their results weren't sensitive to the choice of  $K$  for  $K \geq 3$ . The value of  $K = 3$  remained unchanged for all the results shown throughout.



1

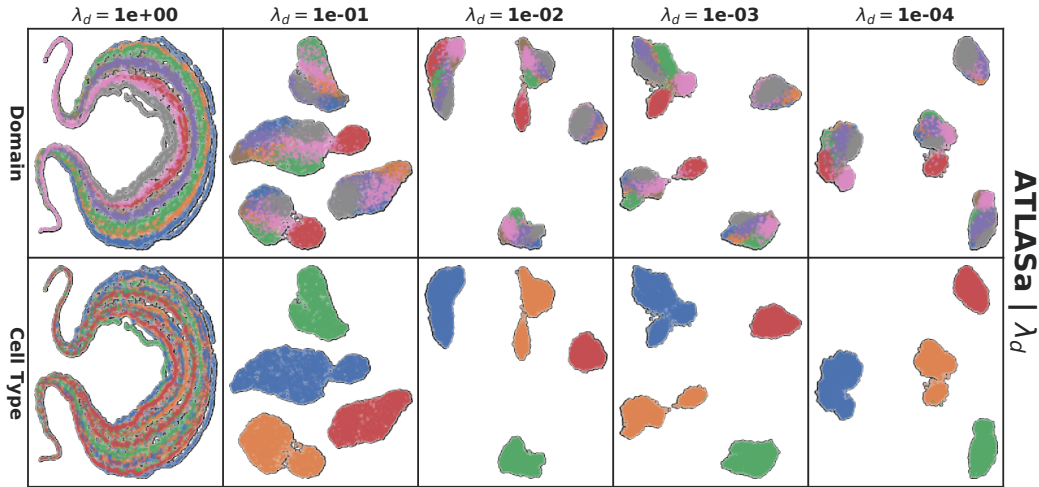


Figure 3: A range of  $\lambda_d$  values for SPLATTER2 with ATLASa

## 5 Performance Evaluation and Benchmarking

In this section, we elaborate on the benchmarking procedure for quantitatively establishing the efficacy of our ATLASa method for learning domain-invariant representations. First, we provide a summary of the external published data-integration methods that we compared our method against (all hyper-parameters for external published data-integration methods were influenced by a recent benchmarking paper [15]) :

- single-cell Variational Inference (scVI) [16]: a correction method that employs a variational autoencoder (VAE) [17] with a Bayesian hierarchical model to embed a gene expression vector, represented by a negative binomial distribution conditioned on batch-labels and measurement-specific variables, to a lower dimensional latent space. scVI was shown to be particularly effective on non-synthetic datasets with intricate effects from domain-misalignment [15]. scVI requires raw-counts, free of any pre-processing, as inputs, which we ensured to be accessible by storing such counts with every dataset. We used scVI Version 0.5.0. Regarding hyper-parameter choices, we trained a negative-binomial based VAE, with two hidden layers in the encoder and decoder, with dimensions 128 and 30, with learning rate .001, for a number of epochs  $\max(400, 8000000 \div \text{number-of-cells})$ , on a single NVIDIA Tesla M60 GPU. Their corrected embedding existed in a 30-dimensional latent space.
- Scanorama: Scanorama [18] is a panorama-sketching inspired integration technique that treats experimental data from different domains as different, smaller "snapshots" of a larger atlas-panorama. It takes these snapshots, aligns different cell types from all different pairs of samples, and "sketches" these snapshots together into a single embeded scRNA-seq "panorama" hyperplane. Scanorama was too shown to be, like scVI, particularly effective on non-synthetic datasets with intricate effects from domain-misalignment [15]. We utilized Version 1.4 of Scanorama with default hyper-parameters.
- Harmony: Harmony [19] is an iterative correction method that takes the principal components of each of the sampled atlases and produces a corrected embedding. Until convergence, Harmony first creates cell clusters of maximum batch-label diversity and then fits an appropriate linear mixture model according to different batches. Harmony was found perform better on synthetic than real biological atlases by [15]. We utilized Version 1.0 of Harmony with default hyper-parameters.

In comparing against these, ATLASa was trained with hyperparameters as defined by the `hp_dict` dictionary in the `tables_12main.ipynb`. The same values for the hyperparameters as specified in the dictionary were used for all the benchmarking analyses.

Now, we elaborate on the metrics we used in the benchmarking section to quantitatively measure the efficacy of ATLASa. All metrics have been adapted to take on values between 0 and 1, with higher values being indicative of better performance for easier interpretation. Furthermore, all metrics were considered on a 'corrected' embedding of the data output by each method ('corrected' refers to the domain-invariant representation of the data learned by each method in this context). We used two metrics, one that evaluated the learned representations on their domain-invariance and their effectiveness is correcting for domain-specific effects and the other metric evaluated the representations on the degree of conservation of biological information :

- K-Nearest-Neighbors Batch Effect Test: kBET [20], is a standard metric for measuring the biases introduced by the fact that the datasets are collected across a vast array of experimental domains. It is used to determine the similarity between the domain-label distribution of cells' nearest neighbors and that of the global atlas. In the original publication, kBET takes on a value between 0 to 1 such that a high kBET value is indicative of more drastic batching effects (in the original publication, higher values were worse). A robust data integration method would thus yield an atlas with a low kBET value. The method first divides the cells into sub-datasets according to cell-type. Then, a kBET value is calculated for each cell type. This is done by running using python's rpy2 library and rpy2.robjests to call the kBET function from R's kBET package. Subsequently, the mean kBET value is found across these cell types. To yield the metric we report as specified above, we subtract said mean from 1 to arrive at a final value for our reported metric and refer to it as kBET in the text. This metric is also used in [6] for evaluating performance on the same task.
- Isolated Label ASW: ILASW, developed by [15], determines how well an integration method accommodates celltypes/labels that are isolated: those not found in every sample. For each isolated label, we first assign, to each datapoint, a binary indicator label with respect to the isolated label. We find the Average Silhouette Width [21], a measure of cluster quality ranging from -1 (intersecting and poorly formed) to 1 (discrete and well formed) with respect to this indicator label using sklearn.metrics's silhouette\_score API and scale it with  $(1 + ASW) \div 2$  to take a value in-between 0 and 1. We lastly take the mean of this over all isolated label names. This describes how well sample-isolated information is retained post-integration. This metric is also used in [6] for evaluating performance on the same task.

## 6 Gene Expression Program Inference for COVID-19 from Healthy and Diseased Tissue Atlases

In constructing Gene Expression Programs for our COVID-19 atlases, we considered three different phenotype labelling regimes:

- Severity: whether a cell is from a healthy patient or one with a severe case of COVID-19.
- Double Positive (DP): whether a cell has non-zero ACE2 and TMPRSS2 expression levels or zero for both.
- Virality: whether a cell has been infected by the virus or is simply a non-infected bystander.

Moreover, we considered multiple different approaches for assigning feature importance measures to construct the gene-expression programs. In all of these approaches, we partitioned the cells according to their cell type. Then, we made a 75:25 train/test split and trained a classifier to predict a phenotype label within each cell type. The top five hundred most important genes were used as the ranked list of genes to define the GEP driving the corresponding phenotype. The classification approaches and corresponding variable importance measures we used were :

- Multi-Layer Perceptron (MLP) with SHAP Values: Training a simple multi-layer perceptron as a classifier. The MLP transformed an input gene expression vector to a 1000 dimensional feature vector using a fully-connected layer, followed by another layer with 100 nodes, followed by the output layer. The classifier was trained using stochastic gradient descent, with the number of epochs ranging between 10 and 16 and batch size between 32 and 256 depending on the training-atlas size (see analysis\_aux.py for further details). We then used the shap package from [22] to derive SHAP value inspired feature importances (see below for a brief explanation of SHAP values and their application in this context).

- We considered both `shap.DeepExplainer` (Deep Shap) and `shap.GradientExplainer` (Grad Shap) (see below again for further explanation).
- In initializing these explainers, we considered two background datasets: a random sample of 1000 training features (which, according to [22], provides a highly accurate estimate of true SHAP values) and the whole dataset. Similarly, for computing the SHAP values themselves, we considered two similar settings: computation using a random sample of 1000 test features or alternatively, using the entire test set. In each setting, we computed the appropriate SHAP values and, following [22], found feature importance by ranking features according to the largest average absolute value of the corresponding entry over the computed SHAP values.
- We found that using the GradientExplainer (Grad Shap), with the background dataset as the whole train-atlas, and computing SHAP values from the whole test-atlas, gave optimal results (as measured by correspondence with independent publications as described in the main text). We thus, used this feature-importance scheme going forwards for this classification approach. The results used in the main text used Grad Shap with these settings due to its efficiency and large observed (REP, PPI, DRUGBANK) intersection sizes on a per-cell-type basis. For Deep Shap, while we used the entire test-atlas to produce SHAP values, we were limited by their implementation to using a 1000-sample of train-atlas gene-expression vectors as a background dataset for the `shap.DeepExplainer` object.
- Random Forest: Using sklearn's [23] `RandomForestClassifier` to classify cells. We set all hyper-parameters to default settings. We computed the Mean Decrease in Impurity for the Random Forest classifier. We used these values from the `feature_importances_` attribute of the classifier in order to determine importances and used them to identify the GEPs.
- Gradient Boosted Decision Trees: Using sklearn's [23] `GradientBoostingClassifier` to classify cells. We set all hyper-parameters to default settings. We again used the `feature_importances_` attribute of the classifier in order to determine importances and used them to identify the GEPs.

### SHAP values and the use of shap with ATLASb

This section contains a brief overview of SHAP (SHapley Additive exPlanations) values [22] and how they are computed in the context of ATLAS. We first consider a machine learning model  $M : \mathbb{R}^n \rightarrow \mathbb{R}$  and a "background" dataset  $B$ ; additionally, let  $e_M = \mathbb{E}_{x \in B} M(x)$ . Given a gene expression vector  $g \in \mathbb{R}^n$ , the corresponding SHAP value of  $g$ ,  $S(g) \in \mathbb{R}^n$ , is one such that  $(\sum_{i=1}^n S(g)_i) + e_M = M(g)$ .  $S(g)$  lets us know how much each feature in  $g$  contributed to the value of  $M(g)$  being different from  $e_M$ . Thus, a SHAP value gives us a way to see how *important* each feature is.

To determine feature importances with the MLP classifier used in ATLASb, we leveraged the `shap` package of Lundberg and Lee [22]. `shap` features an array of model explainers, each with its own `Explainer.shap_values` method that takes in a list of  $m$  gene expression vectors and returns the corresponding SHAP values for each. Each of these explainers takes a model and background dataset as inputs and uses these to in their own computations of SHAP values. We considered two Explainers with our MLP classifier :

- `shap.DeepExplainer`, which approximates SHAP values for neural nets via an extension of DeepLIFT [24].
- `shap.GradientExplainer`, which uses a combination of Integrated Gradients [25], SHAP values, and SmoothGrad [26] as a measure of feature importance.

To evaluate how similar the results from the various classification and feature importance approaches are, we used the full list of all enriched terms returned by the `scanpy.queries.enrich` function when queried on the corresponding GEPs from each of the three methods. Then we compared their similarity using the Szymkiewicz-Simpson coefficient, also known as the overlap coefficient in Tables 1, 2 and 3 below for each feature-importance method used in each class of

phenotype labelling regime. The overlap coefficient for two sets  $X, Y$  is given by  $\frac{|X \cap Y|}{\min(|X|, |Y|)}$ . The Tables 1, 2 and 3 show the overlap coefficient of the GEPs found by other methods when compared to the Grad Shap importance method (for each phenotype labeling scheme and for each cell-type in the labeling scheme). Entries take on values between 0 and 1, with higher values indicating greater similarity between programs.

Tables 4, 5 and 6 describe the size of the intersection between the drug targets found by each of the feature-importance methods used and the (REP,PPI,DRUGBANK) datasets. The entries in the table are shown here as a tuple of (REP intersection size, PPI intersection size, DRUGBANK intersection size).

Finally, Figure 4 contains a Venn diagram showing that the results from all the feature-importance regimes used with ATLASb are remarkably similar lending further credence to the approach and indicating that the drug targets identified may deserve further follow up experiments.

**Table 1: Severity**

<i>Importance</i> \ <i>Cell Type</i>	B	NK	Mast	Macrophages	pDC	Epithelial	mDC	T	Plasma
Grad Shap	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Deep Shap	0.55	0.57	0.77	0.53	0.66	0.67	0.71	0.77	0.89
RF	0.74	0.78	0.54	0.64	0.64	0.61	0.67	0.78	0.70
GBT	0.63	0.85	0.63	0.68	0.54	0.63	0.66	0.71	0.74

**Table 2: Virality**

<i>Importance</i> \ <i>Cell Type</i>	T	Macrophages	Plasma	NK	Epithelial	Neutrophil	...	Epithelial
Grad Shap	1.00	1.00	1.00	1.00	1.00	1.00	...	1.00
Deep Shap	0.52	0.57	0.73	0.66	0.72	0.71	...	0.81
RF	0.53	0.63	0.58	0.69	0.66	0.83	...	0.58
GBT	0.66	0.40	0.39	0.62	0.58	0.65	...	0.65

**Table 3: DP**

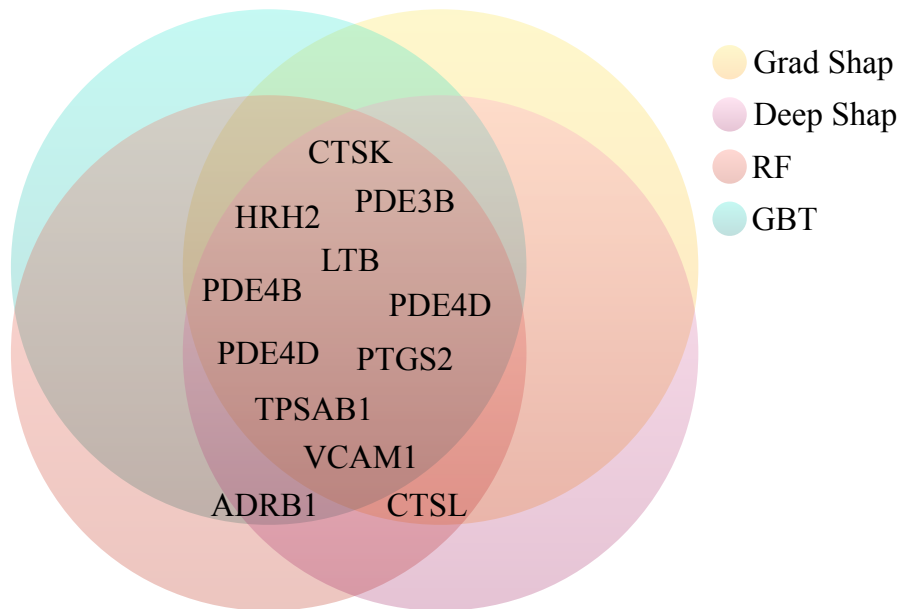
**Table 4: Severity**

<i>Importance</i> \ <i>Cell Type</i>	B	NK	Mast	Macrophages	pDC	Epithelial	mDC	T	Plasma
Grad Shap	(4, 1, 41)	(4, 0, 41)	(2, 5, 31)	(4, 2, 34)	(3, 4, 31)	(2, 4, 26)	(6, 3, 41)	(1, 4, 37)	(1, 2, 30)
Deep Shap	(2, 4, 38)	(4, 1, 42)	(5, 0, 35)	(7, 3, 43)	(2, 5, 34)	(4, 3, 38)	(1, 5, 30)	(4, 4, 31)	(2, 3, 27)
RF	(4, 3, 27)	(2, 4, 35)	(2, 3, 40)	(0, 2, 22)	(5, 2, 31)	(2, 1, 30)	(2, 3, 26)	(1, 5, 33)	(1, 2, 21)
GBT	(1, 2, 27)	(1, 2, 23)	(2, 4, 22)	(3, 0, 31)	(3, 3, 27)	(0, 1, 34)	(2, 4, 31)	(3, 1, 35)	(2, 2, 32)

**Table 5: Virality**

<i>Importance</i> \ <i>Cell Type</i>	T	Macrophages	Plasma	NK	Epithelial	Neutrophil	...	Epithelial
Grad Shap	(3, 3, 38)	(4, 4, 38)	(4, 3, 42)	(3, 0, 24)	(4, 2, 40)	(5, 0, 34)	...	(3, 2, 33)
Deep Shap	(4, 3, 38)	(3, 0, 24)	(4, 2, 37)	(6, 1, 31)	(5, 3, 35)	(4, 4, 38)	...	(3, 1, 31)
RF	(3, 3, 31)	(3, 5, 32)	(3, 4, 38)	(3, 1, 34)	(2, 3, 33)	(3, 0, 24)	...	(3, 1, 28)
GBT	(1, 1, 32)	(3, 0, 24)	(1, 3, 36)	(2, 4, 28)	(4, 1, 37)	(2, 0, 24)	...	(2, 1, 25)

**Table 6: DP**



**Figure 4:** The union, over all phenotype labeling schemes, of drug targets found for all four importance regimes that intersect with those found by the REP study (an independent experimental study on drug re-purposing for COVID-19). We note that all classification/feature-importance methods give an extremely similar results, implying that ATLASb isn't particularly sensitive to the feature importance method used and that the results are robust to that choice.

## 7 Errata and Clarifications

In this section we list typos and errors we spotted in our paper submission between the Paper Submission Deadline and the Supplementary Material Submission Deadline. We apologize for these errors and will make the necessary corrections (along with other errors we identify later in addition to suggested changes by reviewers) in the camera ready version of the paper.

- The legends for Figure 2b in the main text (the first UMAP plot from the left) should say 'Domain' instead of 'Batch'. This error was because 'Batch' and 'Domain' are used interchangeably in the scRNA-seq field, however we use 'Domain' throughout the text so as to not confuse it with 'mini-batch' used to train the model using stochastic gradient descent.
- In the main text, Figures 2a, 2b, 2c, 2d and 3c were not labelled with an appropriately "abcd" panel letter reference. We apologize for this error.
- We apologize for not defining KEGG terms in the main text.

## References

- [1] Kyle J Travaglini, Ahmad N Nabhan, Lolita Penland, Rahul Sinha, Astrid Gillich, Rene V Sit, Stephen Chang, Stephanie D Conley, Yasuo Mori, Jun Seita, Gerald J Berry, Joseph B Shrager, Ross J Metzger, Christin S Kuo, Norma Neff, Irving L Weissman, Stephen R Quake, and Mark A Krasnow. A molecular cell atlas of the human lung from single cell RNA sequencing. August 2019.
- [2] Minoru Kanehisa, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.*, 44(D1):D457–62, January 2016.
- [3] F Alexander Wolf. Alex wolf - Blog/171223\_AnnData\_indexing\_views\_HDF5-backing. [https://falexwolf.de/blog/171223\\_AnnData\\_indexing\\_views\\_HDF5-backing/](https://falexwolf.de/blog/171223_AnnData_indexing_views_HDF5-backing/). Accessed: 2020-6-10.

- [4] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: Simulation of single-cell rna sequencing data. *Genome Biology*, page 174, September 2017.
- [5] Jose Ordovas-Montanes, Daniel F Dwyer, Sarah K Nyquist, Kathleen M Buchheit, Marko Vukovic, Chaarushena Deb, Marc H Wadsworth, 2nd, Travis K Hughes, Samuel W Kazer, Eri Yoshimoto, Katherine N Cahill, Neil Bhattacharyya, Howard R Katz, Bonnie Berger, Tanya M Laidlaw, Joshua A Boyce, Nora A Barrett, and Alex K Shalek. Allergic inflammatory memory in human respiratory epithelial progenitor cells. *Nature*, 560(7720):649–654, August 2018.
- [6] M D Luecken, M Büttner, K Chaichoompu, A Danese, M Interlandi, M F Mueller, D C Strobl, L Zappia, M Dugas, M Colomé-Tatché, and F J Theis. Benchmarking atlas-level data integration in single-cell genomics. May 2020.
- [7] Mingfeng Liao, Yang Liu, Jing Yuan, Yanling Wen, Gang Xu, Juanjuan Zhao, Lin Cheng, Jinxiu Li, Xin Wang, Fuxiang Wang, Lei Liu, Ido Amit, Shuye Zhang, and Zheng Zhang. Single-cell landscape of bronchoalveolar immune cells in patients with covid-19. *Nature Medicine*, page 174, May 2020.
- [8] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, OSDI’16, pages 265–283, USA, November 2016. USENIX Association.
- [9] F Chollet. keras. 2015.
- [10] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.*, 19(1):15, February 2018.
- [11] C S Smillie, M Biton, J Ordovas-Montanes, K M Sullivan, and others. Intra-and inter-cellular rewiring of the human colon during ulcerative colitis. *Cell*, 2019.
- [12] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. May 2015.
- [13] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for Domain-Invariant representation learning. February 2017.
- [14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv [cs.LG]*, December 2014.
- [15] MD Luecken, M Büttner, K Chaichoompu, A Danese, M Interlandi, MF Mueller, DC Strobl, L Zappia, M Dugas, M Colomé-Tatché, and FJ Theis. Benchmarking atlas-level data integration in single-cell genomics. *bioRxiv*, 2020.
- [16] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12):1053–1058, December 2018.
- [17] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.
- [18] Bonnie Berger Brian Hie, Bryan Bryson. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Naure Biotechnology*, 37(June):685–691, 2019.
- [19] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-Ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nat. Methods*, 16(12):1289–1296, December 2019.

- [20] Maren Büttner, Zhichao Miao, F Alexander Wolf, Sarah A Teichmann, and Fabian J Theis. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods*, 16(1):43–49, January 2019.
- [21] Peter Rousseeuw. Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *comput. appl. math.* 20, 53-65. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987.
- [22] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.
- [23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [24] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.
- [25] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *ArXiv*, abs/1703.01365, 2017.
- [26] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *ArXiv*, abs/1706.03825, 2017.

# Insi2vec: A framework for inferring from single-cell and spatial multi-omics

  
US 20220180975A1

(19) **United States**  
(12) **Patent Application Publication** (10) **Pub. No.: US 2022/0180975 A1**  
Regev et al. (43) **Pub. Date: Jun. 9, 2022**

(54) **METHODS AND SYSTEMS FOR DETERMINING GENE EXPRESSION PROFILES AND CELL IDENTITIES FROM MULTI-OMIC IMAGING DATA**

(71) Applicants: **The Broad Institute, Inc.**, Cambridge, MA (US); **Massachusetts Institute of Technology**, Cambridge, MA (US)

(72) Inventors: **Aviv Regev**, Cambridge, MA (US); **Eeshit Dhaval Vaishnav**, Cambridge, MA (US)

(21) Appl. No.: **17/553,691**

(22) Filed: **Dec. 16, 2021**

**Related U.S. Application Data**

(63) Continuation-in-part of application No. 17/426,453, filed on Jul. 28, 2021, filed as application No. PCT/US2020/015481 on Jan. 28, 2020.

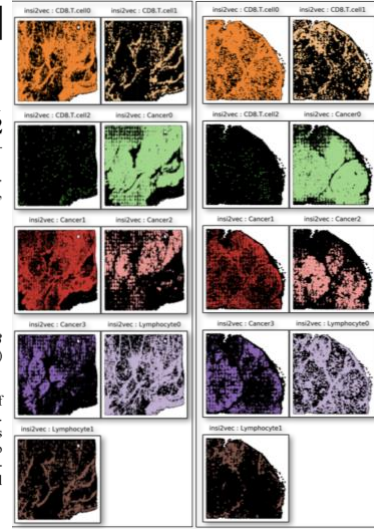
(60) Provisional application No. 62/811,528, filed on Feb. 27, 2019, provisional application No. 62/797,831, filed on Jan. 28, 2019.

**Publication Classification**

(51) **Int. Cl.**  
*G16B 40/30* (2006.01)  
*G16B 25/10* (2006.01)  
*G06N 3/08* (2006.01)

(52) **U.S. Cl.**  
CPC ..... *G16B 40/30* (2019.02); *G06N 3/088* (2013.01); *G16B 25/10* (2019.02)

(57) **ABSTRACT**  
The present disclosure relates to systems and method of determining transcriptomic profile from omics imaging data. The systems and methods train machine learning methods with intrinsic and extrinsic features of a cell and/or tissue to define transcriptomic profiles of the cell and/or tissue. Applicants utilize a convolutional autoencoder to define cell subtypes from images of the cells.  
**Specification includes a Sequence Listing.**



This patent application describes **insi2vec**, *A framework for inferring from single-cell and spatial multi-omics* ([U.S. Patent Application No. 17/553,691](https://patents.google.com/patent/US20220180975A1/en)): *Methods and systems for determining gene expression profiles and cell identities from multi-omic imaging data.* **Contribution:** Co-inventor, with Prof. Aviv Regev.

The patent application for *insi2vec*, describes: (i) a spatio-transcriptomic definition of cell identity using cell intrinsic and cell extrinsic features, and methods for predicting spatial gene expression patterns from (ii) single-cell RNA-sequencing measurements and (iii) histology.

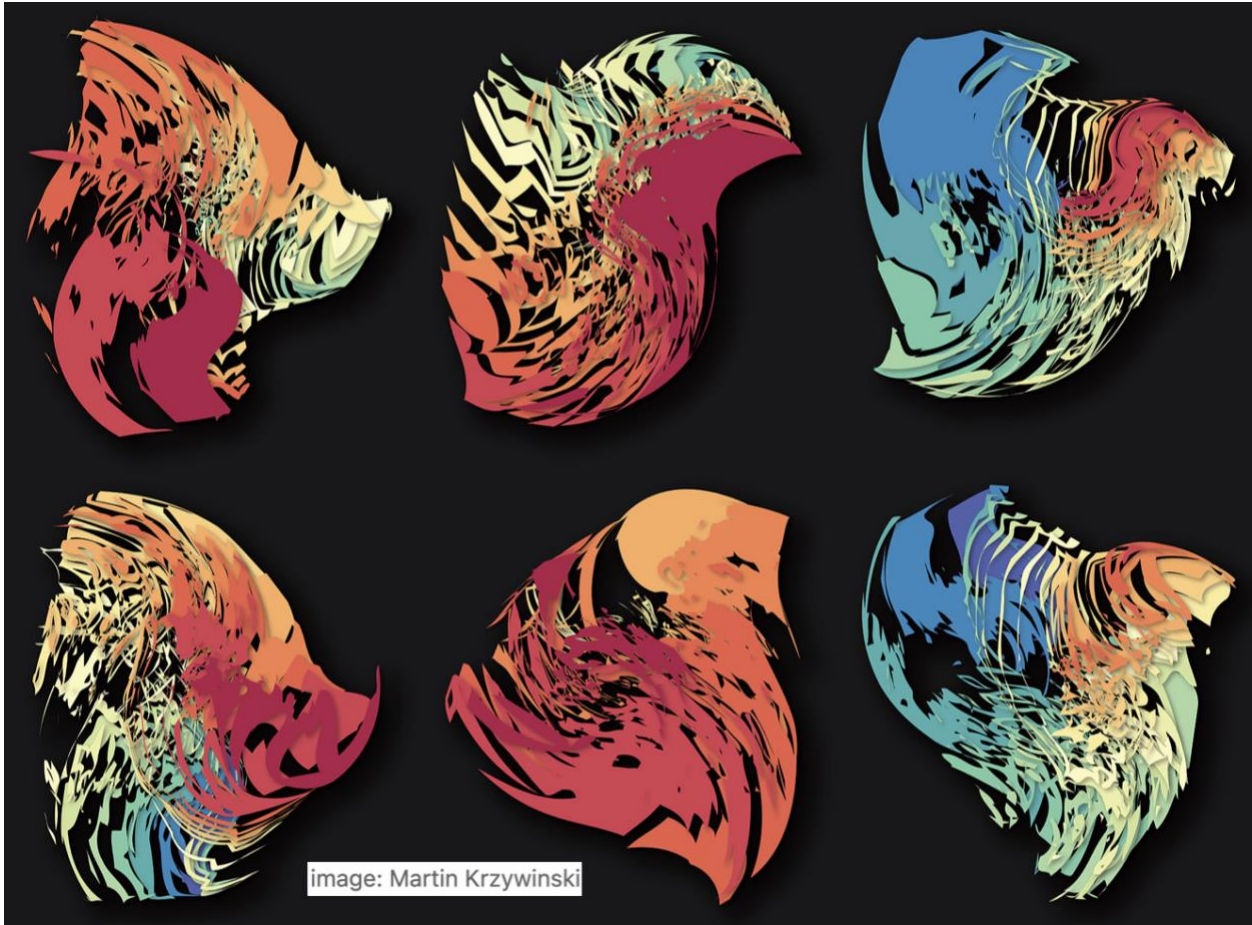
The patent application can be accessed at:

<https://patents.google.com/patent/US20220180975A1/en>.





## Discussion



The thesis described our progress towards building ‘foundation models’ for life science.

In the first part of this thesis, I described a framework for thinking about sequence→function questions. The sequence→function relationship is fascinating. Over the (relatively shorter) time scales that humans tend to think about, sequence→function appears to be the direction of causal flow. However, there is also evidence that the inverse relationship may form a causal loop<sup>1-2</sup>. I like to call this phenomenon ‘mutation-selection entanglement’. There may be more to find here.

Designing a biological sequence with a desired function is fundamental to genome medicine, and over the next decade (and beyond), an enormous class of therapeutic programs and target

discovery strategies will involve the design of (and inference from) biological sequences. A summary of our insights from working towards this goal:

**(i) Robustness and Evolvability.** Not only can we engineer sequences to have the desirable function (like expression levels), we have a framework for engineering them to be robust/evolvable. We think this can be a huge advantage in therapeutics (*eg*: for addressing the short-lived nature of existing sequence-based therapies) and in synthetic biology. As Bianco *et al.*, write in their generous News and Views article<sup>3</sup> about our work: "*...the potential for bioengineering and cellular engineering is enormous. In fact, testing for evolvability can open the door to more stable industrial bioengineering pipelines, as just one example. Mutational robustness shapes the fitness landscape so that fitness peaks are tall but wide instead of narrow, indicating that a broader set of mutations is now buffered and can be tolerated without significant loss of fitness. Selecting for a robust system opens the door, in my opinion, to more efficient production pipelines.*"

**(ii) Modeling.** The accuracy of our sequence→function model was enabled by our unique outlook on modeling the sequence→function relationship: We aren't building a model for simply representing the sequence space. On the contrary, we are modeling the interactions of a sequence with its environment that lead to the function we focus on (e.g.: how a promoter sequence interacts with its environment to generate expression). This is a fundamentally different way of looking at model building, compared to approaches that involve the use of large language model analogs for biological sequence representations where the goal is to model distributions within the sequence space itself. In contrast, our goal isn't just to learn a probability distribution in the

sequence space, without a principled consideration of how these sequences interact with their environment and lead to their function.

**(iii) *Measurement.*** Our approach towards model building is only possible because of the scale and quality of experimental data we have been able to generate. The experimental approaches in our work focus on the measurement of function corresponding to large random samples from the sequence space: in sharp contrast to approaches that focus on mutating/perturbing existing biological systems. For learning accurate sequence→function models, local neighborhoods of existing natural sequences are suboptimal; and our approach of training on de-novo random samples of the sequence space performs significantly better. Additionally, random sequences are exponentially cheaper to synthesize at massive scales (compared to defined sequences), enabling high throughput experimental generation of large-scale labelled sequence-function pair datasets.

**(iv) *Generalizability.*** As long as one operates within the constraints of a system where precise, large-scale sequence→function measurements for random samples from the sequence space are possible, our framework is generalizable across an enormous class of sequence design problems that involve DNA/RNA/peptides. As Bianco *et al* write in their article<sup>3</sup> about our work, *"...Basically, the model learns the space of possible solutions and it is capable of generalizing to a new set of solutions. This is of paramount importance because expression engineering is a fundamental part of industrial bioengineering, especially metabolic engineering. But it is even more important because it allows for uncovering the whole complexity of the organism fitness landscape, which can now be computationally revealed on call."*

Moreover, as Wagner *et al* note in their kind *Nature News & Views* story<sup>4</sup> about our paper, "...And, notably, like other applications of deep learning used in the past few years in biology, such as the development of a tool to predict protein folding [Alphafold], it will enable scientists to answer a broader spectrum of questions than any one group of authors could possibly address."

In this thesis, I also describe the Expression Conservation Coefficient (ECC), which helps detect selection pressures from population genetics data. Existing approaches for identifying signatures of selection on regulatory regions rely on disparate assumptions and data types, and thus (to the best of our knowledge) there currently do not exist appropriate per-gene selection measurements against which we can directly compare the ECC. Below, we contrast the existing body of work in this area, with ECC:

*TF binding (or motif) centric approaches.* One pioneering approach<sup>5</sup> uses mutations within motif-predicted TF binding sites to identify signals of conservation (in *Drosophila*), but without regard for how different TFs impact expression or how TFs interact (this was not possible at the time). This approach was applied to TF(s)-enhancer(s) pairs where the TF was known to play an important role in regulation of the enhancer(s). A similar, more recent study<sup>6</sup> created affinity models based on ChIP-seq data for each TF (in human) and used these models to infer selection from the predicted perturbation of binding. It is unclear how one would generalize such approaches to integrate across all TFs and all regulatory regions, and how their results would compare to the ECC, because each TF would provide a different answer. Earlier approaches<sup>7-8</sup> did not always have TF binding data, but used motifs as surrogates. Nevertheless, the approaches were similar in

principle, in that all of them focus on motifs or TFs but do not integrate across the regulatory sequence. This is a key difference from the ECC because, as we have shown in previously<sup>9</sup>, and others have predicted<sup>10</sup>, strong scoring motifs only account for a portion of a gene's regulation.

*Reporter based approaches.* Other methods<sup>11</sup> use the measured effects of mutations within regulatory regions assayed in a reporter assay (in human). Consequently, these can only be applied to regulatory regions for which mutation effects have been experimentally determined, which is not available for the vast majority of sequences one would encounter.

*Mutation counting based approaches.* Methods that are based on counting mutations within a regulatory region (or inter-species comparisons of regulatory sequence) often require a background set of sequences that are assumed to be neutral against which to compare<sup>12-15</sup>. However, in a genome as compact as *S. cerevisiae*'s, it is not clear if there are sufficient locations that are non-functional. Furthermore, these methods assume that the mutation rate is uniform across the genome, which is unlikely to be the case. Finally, we did show that sequence divergence within promoter sequences (*i.e.*, the numerator in such methods) does not correlate with other measures of selection (**Extended Data Fig. 4c**), arguing that such methods would not fare well in a comparison with the ECC.

In the second part of this thesis, I proposed frameworks for thinking about gene expression. Expression lends itself beautifully to the study of genotype→phenotype→fitness. If there is a phenotype that is better suited to this line of scientific inquiry than gene expression, I haven't

found it yet! This part of the thesis focused on applications of these frameworks to cancer, brain and COVID-19 research.

With ATLAS, we introduced a novel framework for predicting and prioritizing putative human disease drug targets from single-cell gene expression measurements. As with all hypothesis generating machine learning methods, one must be extremely careful when interpreting these results and treating them as definitive. For instance, the COVID-19 REP dataset that we used for one of the validations shown above was generated using Vero E6 cells, which are monkey kidney epithelial cell lines. These in-vitro results may not necessarily translate to complex real world biological systems. Even though our intersections with drug targets found from other independent sources is a promising and interesting result, these results must be looked at with extreme caution and require comprehensive further validation using experimental assays and multi-stage rigorous clinical trials before entering the clinic. It is encouraging, though, that our results align with observations made from completely independent ways of analyzing COVID-19 in the context of putative drug targets from scRNA-seq, PPI and drug re-purposing studies. Protein and RNA levels are often known to be uncorrelated in biological systems, and the fact that our results hold across these domains is quite promising. We would also like to emphasize that while the results from our approach, like from any hypothesis generating approach, may provide very interesting and promising putative drug targets, it is imperative that these be validated experimentally and put through rigorous rounds of evaluation. This is an essential step to before any of these results may be considered for real world applications.

We expect utility of feature importance metrics, such as SHAP values, for non-linear and black-box models, in order to identify and characterize complex biological phenomena, to broadly impact the life sciences. We hope that the early demonstration of results using these simple models and variable importance metrics here spurs exponential developments in better approaches for studying a wide range of diseases and biological phenomena. As single cell data is being generated at an ever-faster pace, traditional methods face massive challenges with this unprecedented scale of data. ATLAS has thus been developed with an emphasis on scalability and efficiency.

In summary, I see neural networks' (and more generally, machine learning's) important role in the design→build→test→learn cycle as a starting point. There are innumerable intriguing open questions, downstream of (and orthogonal to) this cycle. Inductive application of principles over (evolutionary) time and (sequence) space, as introduced here<sup>16</sup>, will lead us to interesting answers to some of these questions.

Thank you for reading!



## References

1. Monroe, J.G., Srikant, T., Carbonell-Bejerano, P. *et al.* Mutation bias reflects natural selection in *Arabidopsis thaliana*. *Nature* **602**, 101–105 (2022).
2. Burgess, D.J. Tuning mutagenesis by functional outcome. *Nat Rev Genet* **23**, 135 (2022).
3. Bianco, S. Artificial Intelligence: Bioengineers' Ultimate Best Friend. *GEN Biotechnology*. Apr 2022. 140-141.
4. Wagner. A. AI predicts the effectiveness and evolution of gene promoter sequences. *Nature News and Views*. March 2022.
5. Moses, Alan M. 2009. “Statistical Tests for Natural Selection on Regulatory Regions Based on the Strength of Transcription Factor Binding Sites.” *BMC Evolutionary Biology* 9 (December): 286.
6. Liu, Jialin, and Marc Robinson-Rechavi. 2020. “Robust Inference of Positive Selection on Regulatory Sequences in the Human Brain.” *Science Advances* 6 (48).
7. Gasch, Audrey P., Alan M. Moses, Derek Y. Chiang, Hunter B. Fraser, Mark Berardini, and Michael B. Eisen. 2004. “Conservation and Evolution of Cis-Regulatory Systems in Ascomycete Fungi.” *PLoS Biology* 2 (12): e398.
8. Habib, Naomi, Ilan Wapinski, Hanah Margalit, Aviv Regev, and Nir Friedman. 2012. “A Functional Selection Model Explains Evolutionary Robustness despite Plasticity in Regulatory Networks.” *Molecular Systems Biology* 8: 619.
9. Boer, Carl G. de, Eeshit Dhaval Vaishnav, Ronen Sadeh, Esteban Luis Abeyta, Nir Friedman, and Aviv Regev. 2020. “Deciphering Eukaryotic Gene-Regulatory Logic with 100 Million Random Promoters.” *Nature Biotechnology* 38 (1): 56–65.
10. Tanay, Amos. 2006. “Extensive Low-Affinity Transcriptional Interactions in the Yeast Genome.” *Genome Research* 16 (8): 962–72.
11. Smith, Justin D., Kimberly F. McManus, and Hunter B. Fraser. 2013. “A Novel Test for Selection on Cis-Regulatory Elements Reveals Positive and Negative Selection Acting on Mammalian Transcriptional Enhancers.” *Molecular Biology and Evolution* 30 (11): 2509–18.
12. Haygood, Ralph, Olivier Fedrigo, Brian Hanson, Ken-Daigoro Yokoyama, and Gregory A. Wray. 2007. “Promoter Regions of Many Neural- and Nutrition-Related Genes Have Experienced Positive Selection during Human Evolution.” *Nature Genetics* 39 (9): 1140–44.
13. McDonald, J. H., and M. Kreitman. 1991. “Adaptive Protein Evolution at the Adh Locus in *Drosophila*.” *Nature* 351 (6328): 652–54.
14. Andolfatto, Peter. 2005. “Adaptive Evolution of Non-Coding DNA in *Drosophila*.” *Nature* 437 (7062): 1149–52.
15. Hahn, Matthew W. 2007. “Detecting Natural Selection on Cis-Regulatory DNA.” *Genetica* 129 (1): 7–18.
16. Vaishnav, E.D., de Boer, C.G., Molinet, J. *et al.* The evolution, evolvability and engineering of gene regulatory DNA. *Nature* **603**, 455–463 (2022).

# Evolution, Evolvability, Expression and Engineering

by

Eeshit Dhaval Vaishnav

Bachelor of Technology, Indian Institute of Technology Kanpur

Submitted to the Department of Biology in Partial Fulfillment  
of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

SEPTEMBER 2022

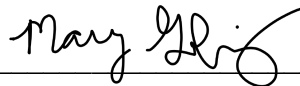
© 2022 MIT. All rights reserved.



Signature of the Author: \_\_\_\_\_ August 31, 2022



Certified by: \_\_\_\_\_ Prof. Aviv Regev  
Professor of Biology  
Thesis Supervisor



Accepted by: \_\_\_\_\_ Prof. Mary Gehring  
Member, Whitehead Institute  
Director, Biology Graduate Committee

# Summary of Work

## PUBLICATIONS

- **Nature**: [1] (First (and a corresponding) author) ([Nature cover](#)), [2] ([Nature cover](#)), [3], [4]
- **Nature Medicine**: [5] (co-first author)
- **Nature Biotechnology**: [6]
- **Nature Communications**: [7]
- **Cell**: [8]
- **bioRxiv**: [9]

[1] [The evolution, evolvability and engineering of gene regulatory DNA](#)

[2] [A multimodal cell census and atlas of the mammalian primary motor cortex](#)

[3] [A transcriptomic and epigenomic cell atlas of the mouse primary motor cortex](#)

[4] [The human body at cellular resolution: the NIH Human Biomolecular Atlas Program](#)

[5] [Single-cell meta-analysis of SARS-CoV-2 entry genes across tissues and demographics](#)

[6] [Deciphering eukaryotic gene-regulatory logic with 100 million random promoters](#)

[7] [Actomyosin meshwork mechanosensing enables tissue shape to orient cell force](#)

[8] [A Cellular Taxonomy of the Bone Marrow Stroma in Homeostasis and Leukemia](#)

[9] [Reference-based cell type matching of spatial transcriptomics data](#)

**Google Scholar**: <https://scholar.google.com/citations?user=brVs5bAAAAAJ&hl=en>

## PATENT

**insi2vec**, *A framework for inferring from single-cell and spatial multi-omics* ([U.S. Patent Application No. 17/553,691](#)): Methods and systems for determining gene expression profiles and cell identities from multi-omic imaging data

# Acknowledgements

I would like to express my most sincere gratitude to Prof. Aviv Regev (who is the best PhD advisor on the planet), MIT and the Broad Institute for the unbounded opportunities. I would also like to thank them, and my collaborators, thesis committee, colleagues, friends and family for their unwavering support and kindness. The work presented here would be impossible without all of their contributions.

# Evolution, Evolvability, Expression and Engineering

by

Eeshit Dhaval Vaishnav

Submitted on August 31, 2022 in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy at the Massachusetts Institute of Technology

## ABSTRACT

This thesis describes how to build machines (*Engineering*) that answer questions about: (a) *Evolution & Evolvability* and (b) *Expression*.

In the first part of this thesis, I present a framework for understanding and engineering biological sequences, and solving sequence→function problems by building ‘Complete Fitness Landscapes’ in sequence space. This framework for measuring, modelling and designing biological sequences is built around the idea of learning an ‘oracle’ (typically a deep neural network model that takes a sequence as input and predicts its corresponding function) to traverse these ‘Complete Fitness Landscapes’. Here we develop a (promoter sequence)→(gene expression) oracle and use it with our framework to design sequences that demonstrate expression beyond the range of naturally observed sequences. We also show how our framework can be used to detect signatures of selection on a sequence, and to characterize robustness and evolvability.

The second part of this thesis describes two frameworks for inferring from single-cell and spatial gene expression measurements: ATLAS (A Tool for Learning from Atlas-scale Single-cell datasets) and insi2vec (a framework for inferring from spatial multi-omic and imaging measurements).

Thesis Supervisor: Prof. Aviv Regev  
Title: Professor of Biology, MIT  
Core Member, Broad Institute  
Investigator, Howard Hughes Medical Institute  
Head, Roche Genentech Research and Early Development.

# Table of Contents

**Title**.....1

**Summary**.....2

**Acknowledgements** .....3

**Abstract**.....4

**Introduction**.....6

**Part A: *Evolution & Evolvability***.....20

**Part B: *Expression*** .....134

**Discussion**.....159

# Introduction

The goals of the work described in this thesis were to (i) find important questions, (ii) build machines for answering such questions and (iii) construct frameworks for discovering new knowledge. Each of these remain works in progress.

Discovering new knowledge (iii) is almost certainly the most challenging of these goals. This may be because new knowledge seems to emerge spontaneously, almost as a by-product of working on finding important questions and answering them. Reliably conjuring new knowledge likely requires that we build general purpose machines for answering questions.

The significance of finding (and choosing) important questions (i) cannot be overstated. Fortunately, in science, there is a way of doing this reproducibly. If a scientist reads sufficient scientific literature in their areas of interest, the important questions will often become abundantly clear to them. A useful test for a scientist to assess whether a question is ‘important’, is to ask whether the ‘importance’ of these questions becomes immediately clear to everyone else they communicate these questions to. E.g.:

- *Can we predict evolution?*
- *Can we predict gene expression?*
- *Could we discover emergent phenomena from genomes of populations?*
- *Could we understand evolutionary history?*
- *Can we predict future evolvability?*
- *Could we discover principles that, when inductively applied over (evolutionary) time and (sequence) space answer, these questions?*

When working on such audacious grand questions, it can be useful to think of them as ‘homework problems’. This helps convince the scientist that an answer exists, which reflexively helps the process of finding answers and discovering new knowledge. More generally, ‘working backwards from the homework assignments in existing literature and review papers’ can be a useful starting point for thinking about scientific questions.

Once the questions are defined, the work of building machines for answering them (ii) begins. Neural networks are the closest humans have come, to building a general purpose technology for answering questions. I don’t think it is a coincidence (or simply a result of the time period the work presented here was carried out in) that neural networks, which are universal function approximators, ended up becoming pivotal to the work presented in this thesis. Apart from the critical role neural network models played in the design→build→test→learn cycles described here, they were also instrumental in their role as ‘foundation models’ for tasks downstream of this cycle. Much of the work presented here involved the formulation of questions in a way that allowed these machines to learn how to answer the questions posed to them (*machine learning*). Machine learning thus became a critical tool in the process of finding new knowledge (*research*).

This thesis describes our research on building machines (*Engineering*) to answer questions about: (a) *Evolution & Evolvability* and (b) *Expression*.



## EVOLUTION & EVOLVABILITY

In the first part of this thesis, I discuss a framework for thinking about sequence→function problems in terms of ‘Complete Fitness Landscapes’ in sequence space. This framework for measuring, modelling and designing biological sequences is built around the idea of learning an ‘oracle’ (typically a deep neural network model that takes a sequence as input and predicts its corresponding function) to traverse these ‘Complete Fitness Landscapes’. Here we develop a (promoter sequence)→(gene expression) oracle and use it with our framework to design sequences that demonstrated expression beyond the range of naturally observed sequences. We also show how our framework can be used to detect signatures of selection on a sequence, and to characterize robustness and evolvability.

Non-coding regulatory DNA sequences regulate the expression of protein coding sequences of a gene. Changes in regulatory DNA play a major role in the evolution of gene expression<sup>1</sup>. Mutations in *cis*-regulatory elements (CREs) can affect their interactions with transcription factors (TFs), change the timing, location, and level of gene expression, and impact organismal phenotype and fitness<sup>2,3</sup>. While TFs evolve slowly because they each regulate many target genes, CREs evolve much faster and are thought to drive substantial phenotypic variation<sup>7</sup>. Thus, understanding how *cis*-regulatory sequence variation affects gene expression, phenotype and organismal fitness is fundamental to our understanding of regulatory evolution<sup>2</sup>.

A fitness function maps genotypes (which vary through mutations) to their corresponding organismal fitness values (where selection operates)<sup>8</sup>. A complete fitness landscape<sup>9</sup> is defined by a fitness function that maps each sequence in a sequence space to its associated fitness, coupled with an approach for visualizing the sequence space. Partial fitness landscapes have been characterized empirically<sup>4,5,10</sup>, often defining fitness as the maximum growth rate of single-cell

organisms<sup>4,11</sup>. Many recent empirical fitness landscape studies of proteins<sup>12</sup>, adeno-associated viruses<sup>13</sup>, catalytic RNAs<sup>14</sup>, promoters<sup>15</sup>, and TF binding sites<sup>16</sup> have favored molecular activities as fitness proxies because they are less susceptible to experimental biases and measurement noise<sup>17</sup>. In particular, the molecular activity of a promoter sequence as reflected in the expression of the regulated gene has been used to build a ‘promoter fitness landscape’<sup>18</sup>. However, despite advances in high-throughput measurements, empirical fitness landscape studies often sample sequences in the local neighborhood of natural ones and thus remain limited to a tiny subset of the complete sequence space whose size grows exponentially with sequence length ( $4^L$  for DNA or RNA, where  $L$  is the length of sequence)<sup>4-6</sup>.

Understanding the relationship between promoter sequence, expression phenotype, and fitness would allow us to answer fundamental questions<sup>6</sup> in evolution and gene regulation, and provide an invaluable bioengineering tool<sup>6,19</sup>. A model that accurately approximates the relationship between sequence and expression can serve as an “oracle” in evolutionary studies to conduct and interpret *in-silico* experiments<sup>20-23</sup>, predict which regulatory mutations affect expression and fitness (when coupled with expression-to-fitness curves<sup>11</sup>), design or evolve new sequences with desired characteristics, determine how quickly selection achieves an expression optimum, identify signatures of selective pressures on extant regulatory sequences, visualize fitness landscapes and characterize mutational robustness and evolvability<sup>2,4-6,24,25</sup>.

In the first part of this this thesis, we address these long-standing problems by developing a framework for studying regulatory evolution and fitness landscapes based on *Saccharomyces cerevisiae* promoter sequence-to-expression models.

## EXPRESSION

The second part of this thesis focusses on approaches for inferring from single-cell and spatial gene expression measurements. Single-cell RNA-sequencing measurements (scRNA-seq)<sup>26</sup> output a ‘feature vector’ for each cell corresponding the cell’s gene expression profile (referred to as its transcriptome). We first describe an approach, that we call ATLAS (A Tool for Learning from Atlas-scale Single-cell datasets), for inferring from large datasets of scRNA-seq measurements (referred to as scRNA-seq atlases). ATLAS allows us to integrate vast scRNA-seq datasets, decipher gene expression programs, predict and prioritize drug targets. ATLAS can also spatially map cell types and predict spatial gene expression patterns. We demonstrate the applications of ATLAS in the context of COVID-19<sup>27</sup>.

***ATLAS:*** The Coronavirus Disease 2019 (COVID-19) caused by the SARS-CoV-2 virus has led to significant global morbidity and mortality<sup>28</sup>. With no standard cures or vaccines available yet to contain and prevent the disease, there is an emergent need to understand COVID-19 onset, progression and the underlying disease mechanisms at the molecular level to enable discovery of pathways and targets for treatments. COVID-19 starts with the infection or entry of SARS-CoV-2 into human cells, primarily in the upper respiratory system. Subsequent replication and spread of the virus to other human cells, mounts inflammation and immune responses leading to the widespread clinical pathologies including pneumonia in moderate cases, and acute respiratory distress and death in severe cases. A key to understanding disease mechanisms in turn lies in identifying molecular changes in virus infected cells observed in COVID-19 patients. The genetic code in each of these cells is organized into (over 20000) units called genes, which are converted into functional gene products like proteins through a process referred to as gene

expression. Gene expression is regulated by gene regulatory networks that involve complex, non-linear and combinatorial interactions between genes<sup>29</sup>. Human phenotypes, such as "healthy" or "diseased" states, are induced by the expression of these genes in relevant virus-infected and bystander responsive cells. Single-cell RNA-sequencing (scRNA-seq) is an experimental technology that allows us to measure gene expression in such individual cells isolated from tissues. The output of every scRNA-seq experiment is a high-dimensional, sparse gene expression vector for each isolated cell whose dimensions correspond to the expression levels of genes within the cell. Machine learning methods are then applied to the output of scRNA-seq experiments for dimensionality reduction<sup>30</sup>, unsupervised clustering<sup>31</sup> and gene expression program inference<sup>32</sup>. The construction of single-cell atlases involves multiple scRNA-seq experiments run on tissues isolated from a diverse set of individuals, who are dispersed along a continuum of phenotypes. These experiments are conducted under varying experimental conditions and run on a wide range of technological platforms in different laboratories. This introduces complex, non-linear variability in measurements made across these experimental domains and hinders our ability to make effective comparisons between measurements made across domains (e.g. comparing differences in gene expression programs between healthy and diseased individuals). These domain-specific effects are an impediment to the process of making fundamental biological discoveries and to translational applications such as the identification of targets for therapeutic interventions to treat diseases like COVID-19. Extensive previous work on the identification of drug targets has focused on convolutional and graph neural networks for modeling protein structures<sup>33</sup> and molecular interactions<sup>34-35</sup> between drugs and their putative targets. But, a critical upstream step in drug development is the identification of these biological

targets. Single-cell atlases have the potential to transform the process of identification of genes involved in the modulation of disease phenotype.

There is currently no principled, generalizable approach for identifying drug targets by inferring gene expression programs from single-cell atlases to treat human diseases. To address this, one would need to adequately address the two-fold challenge in single-cell atlas analysis described above : (i) the combinatorial and non-linear effects of gene expression on phenotypes and (ii) the variability in measurements across experimental domains.

The task of remedying the variability in scRNA-seq measurements across experimental domains is a type of domain adaptation<sup>36</sup> problem where the objective is to learn domain-invariant feature representations of scRNA-seq data. Domain-invariant feature representations are central to a large body of work on domain adaptation<sup>37-41</sup>. In the context of scRNA-seq, domain adaptation is referred to as data-integration and is defined as the process of generating an internally- consistent version of the data<sup>42</sup> across these measurement domains. Existing methods carry out this scRNA-seq data-integration task by either operating directly on the full gene expression vectors<sup>43-44</sup> or operating on representations derived from them like nearest-neighbor graphs<sup>45-47</sup> and learned low-dimensional embeddings<sup>48-51</sup>. These methods employ a broad range of statistical and machine learning techniques including panoramic stitching<sup>52</sup>, canonical correlation analysis<sup>53</sup>, non-negative matrix factorization<sup>54</sup> and perturbation modeling<sup>49</sup>. Consequently, comprehensive metrics<sup>55</sup> have been developed for evaluating the efficacy of these domain integration methods in ameliorating domain-specific effects while conserving biological information. Benchmarking studies<sup>42,56</sup> for the domain integration task using these metrics demonstrate the need and room for vast improvements.

Additionally, the domain-invariant representations learned by many of these methods<sup>46,54,57-58</sup> cannot be mapped back to the input gene expression vectors which significantly restricts the applicability of such representations to biologically meaningful tasks.

On the other hand, an autoencoder (AE)<sup>59-60</sup> has the ability to learn feature representations that can be mapped back to the input space using a pair of functions: an encoder for transforming the input into this representation and a decoder for reconstructing the input from this representation. AEs can learn domain-invariant representations by minimizing the domain discrepancy between features learned by the encoder using a domain regularizer or by encouraging domain confusion using an adversarial objective<sup>41,61</sup>. Maximum Mean Discrepancy (MMD)<sup>62</sup>, a widely used regularizer for aligning the first two moments of a distribution, has recently been applied to the scRNA-seq data- integration problem<sup>49-51</sup>. However, scRNA-seq data<sup>63</sup> and their latent representations learned by an AE follow non-Normal distributions. First and second order moment matching methods do not suffice for minimizing the domain discrepancy in many real-world unsupervised domain adaptation problems and so methods for matching higher-order statistics have been proposed<sup>64-66</sup> for the task of transferring labels from a labelled source domain to an unlabelled target domain. However, an approach for incorporating higher order moment-matching regularizers to the multi-target domain adaptation<sup>67</sup> task in general, and the scRNA-seq data integration problem, in particular is lacking.

We hypothesize that domain-invariant feature representations of scRNA-seq data can help address the pressing need for identification of human disease drug targets when used in tandem with

modern feature importance methods to account for the combinatorial and non-linear effects of gene-expression on phenotype. This forms the basis of ATLAS.

*insi2vec*: scRNA-seq captures cell-intrinsic information but the process of making these measurements, which involves the dissociation of tissues, leads to a loss of spatial and cell-extrinsic contextual information. Spatial transcriptomic measurements overcome these limitations, but come with their own sets of trade-offs<sup>68</sup>. We present *insi2vec*<sup>69</sup>, a framework for inferring from single-cell and spatial multi-omics, to address these challenges. *insi2vec*, consists of: (i) a spatio-transcriptomic definition of cell identity using cell intrinsic and cell extrinsic features, and (ii) methods for predicting spatial gene expression patterns from (ii-a) single-cell RNA-sequencing measurements and (ii-b) histology. We demonstrate the applications of *insi2vec* to cancer and brain research.

## References

1. Wittkopp, P. J. & Kalay, G. Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence. *Nat. Rev. Genet.* **13**, 59–69 (2011).
2. Hill, M. S., Vande Zande, P. & Wittkopp, P. J. Molecular and evolutionary processes generating variation in gene expression. *Nature Reviews Genetics* 1–13 (2020).
3. Fuqua, T. *et al.* Dense and pleiotropic regulatory information in a developmental enhancer. *Nature* 1–5 (2020).
4. de Visser, J. A. G. M. & Krug, J. Empirical fitness landscapes and the predictability of evolution. *Nat. Rev. Genet.* **15**, 480–490 (2014).
5. Kondrashov, D. A. & Kondrashov, F. A. Topological features of rugged fitness landscapes in sequence space. *Trends in Genetics* **31**, 24–33 (2015).
6. de Visser, J. A. G. M., Elena, S. F., Fragata, I. & Matuszewski, S. The utility of fitness landscapes and big data for predicting evolution. *Heredity (Edinb)* **121**, 401–405 (2018).
7. Weirauch, M. T. & Hughes, T. R. Conserved expression without conserved regulatory sequence: the more things change, the more they stay the same. *Trends Genet.* **26**, 66–74 (2010).
8. Orr, H. A. The genetic theory of adaptation: a brief history. *Nature Reviews Genetics* **6**, 119–127 (2005).
9. Weinreich, D. M., Lan, Y., Wylie, C. S. & Heckendorn, R. B. Should evolutionary geneticists worry about higher-order epistasis? *Current Opinion in Genetics & Development* **23**, 700–707 (2013).
10. Venkataram, S. *et al.* Development of a Comprehensive Genotype-to-Fitness Map of Adaptation-Driving Mutations in Yeast. *Cell* **166**, 1585-1596.e22 (2016).
11. Keren, L. *et al.* Massively Parallel Interrogation of the Effects of Gene Expression Levels on Fitness. *Cell* **166**, 1282-1294.e18 (2016).
12. Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature* **533**, 397–401 (2016).
13. Ogden, P. J., Kelsic, E. D., Sinai, S. & Church, G. M. Comprehensive AAV capsid fitness landscape reveals a viral gene and enables machine-guided design. *Science* **366**, 1139–1143 (2019).
14. Pitt, J. N. & Ferré-D’Amaré, A. R. Rapid construction of empirical RNA fitness landscapes. *Science* **330**, 376–379 (2010).
15. Shultzaberger, R. K., Malashock, D. S., Kirsch, J. F. & Eisen, M. B. The Fitness Landscapes of cis-Acting Binding Sites in Different Promoter and Environmental Contexts. *PLOS Genetics* **6**, e1001042 (2010).
16. Mustonen, V., Kinney, J., Callan, C. G. & Lässig, M. Energy-dependent fitness: a quantitative model for the evolution of yeast transcription factor binding sites. *Proc. Natl. Acad. Sci. U.S.A.* **105**, 12376–12381 (2008).
17. Hartl, D. L. What Can We Learn From Fitness Landscapes? *Curr Opin Microbiol* **0**, 51–57 (2014).



18. Otwinowski, J. & Nemenman, I. Genotype to phenotype mapping and the fitness landscape of the *E. coli* lac promoter. *PLoS ONE* **8**, e61570 (2013).
19. Sinai, S. & Kelsic, E. D. A primer on model-guided exploration of fitness landscapes for biological sequence design. *arXiv:2010.10614 [cs, q-bio]* (2020).
20. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods* **12**, 931–934 (2015).
21. Avsec, Ž. *et al.* Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat. Genet.* **53**, 354–366 (2021).
22. Shrikumar, A., Greenside, P. & Kundaje, A. Learning Important Features Through Propagating Activation Differences. *arXiv [cs.CV]* (2017).
23. Avsec, Ž. *et al.* Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* **18**, 1196–1203 (2021).
24. Fragata, I., Blanckaert, A., Louro, M. A. D., Liberles, D. A. & Bank, C. Evolution in the light of fitness landscape theory. *Trends in Ecology & Evolution* **34**, 69–82 (2019).
25. Payne, J. L. & Wagner, A. The causes of evolvability and their evolution. *Nature Reviews Genetics* **20**, 24–38 (2019).
26. Tanay, A., Regev, A. Scaling single-cell genomics from phenomenology to mechanism. *Nature* **541**, 331–338 (2017).
27. Muus, C., Luecken, M.D., Eraslan, G. *et al.* Single-cell meta-analysis of SARS-CoV-2 entry genes across tissues and demographics. *Nat Med* **27**, 546–559 (2021).
28. Mortality analyses. <https://coronavirus.jhu.edu/data/mortality>. Accessed: 2020-6-4.
29. Ido Amit, Manuel Garber, Nicolas Chevrier, Ana Paula Leite, Yoni Donner, Thomas Eisenhaure, Mitchell Guttman, Jennifer K Grenier, Weibo Li, Or Zuk, Lisa A Schubert, Brian Birditt, Tal Shay, Alon Goren, Xiaolan Zhang, Zachary Smith, Raquel Deering, Rebecca C McDonald, Moran Cabili, Bradley E Bernstein, John L Rinn, Alex Meissner, David E Root, Nir Hacohen, and Aviv Regev. Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses. *Science*, 326(5950):257–263, October 2009.
30. Pierson, E., Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol* **16**, 241 (2015).
31. Wang, B., Zhu, J., Pierson, E. *et al.* Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat Methods* **14**, 414–416 (2017).
32. Dylan Kotliar, Adrian Veres, M Aurel Nagy, Shervin Tabrizi, Eran Hodis, Douglas A Melton, and Pardis C Sabeti. Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq. *Elife*, 8, July 2019.
33. Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Židek, Alexander W R Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, January 2020.

34. Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. March 2017.
35. Joseph Gomes, Bharath Ramsundar, Evan N Feinberg, and Vijay S Pande. Atomic convolutional networks for predicting Protein-Ligand binding affinity. March 2017.
36. Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1):151–175, May 2010.
37. Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. January 2013.
38. Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, February 2013.
39. Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. January 2019.
40. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for Large-Scale sentiment classification: A deep learning approach. January 2011.
41. Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.
42. M D Luecken, M Büttner, K Chaichoompu, A Danese, M Interlandi, M F Mueller, D C Strobl, L Zappia, M Dugas, M Colomé-Tatché, and F J Theis. Benchmarking atlas-level data integration in single-cell genomics. May 2020.
43. W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, January 2007.
44. Uri Shaham, Kelly P Stanton, Jun Zhao, Huamin Li, Khadir Raddassi, Ruth Montgomery, and Yuval Kluger. Removal of batch effects using distribution-matching residual networks. *Bioinformatics*, 33(16):2539–2546, August 2017.
45. Krzysztof Polan’ski, Matthew D Young, Zhichao Miao, Kerstin B Meyer, Sarah A Teichmann, and Jong-Eun Park. BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics*, 36(3):964–965, 08 2019.
46. Nikolas Barkas, Viktor Petukhov, Daria Nikolaeva, Yaroslav Lozinsky, Samuel Demharter, Konstantin Khodosevich, and Peter V Kharchenko. Joint analysis of heterogeneous single-cell RNA-seq dataset collections. *Nat. Methods*, 16(8):695–698, August 2019.
47. Laleh Haghverdi, Aaron T L Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.*, 36(5):421–427, June 2018.
48. Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12):1053–1058, December 2018.

49. Mohammad Lotfollahi, Mohsen Naghipourfar, Fabian Theis, and F. Wolf. Conditional out-of-sample generation for unpaired data using trvae. *arXiv*, 10 2019.
50. Valentine Svensson, Adam Gayoso, Nir Yosef, and Lior Pachter. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics*, March 2020.
51. Matthew Amodio, David van Dijk, Krishnan Srinivasan, William S Chen, Hussein Mohsen, Kevin R Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy, Anita Desai, V Ravi, Priti Kumar, Ruth Montgomery, Guy Wolf, and Smita Krishnaswamy. Exploring single-cell data with deep multitasking neural networks. *Nat. Methods*, 16(11):1139–1145, November 2019.
52. Bonnie Berger Brian Hie, Bryan Bryson. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature Biotechnology*, 37(June):685–691, 2019.
53. Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck, 3rd, Yuhao Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of Single-Cell data. *Cell*, 177(7):1888–1902.e21, June 2019.
54. Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-Cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887.e17, June 2019.
55. Maren Büttner, Zhichao Miao, F Alexander Wolf, Sarah A Teichmann, and Fabian J Theis. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods*, 16(1):43–49, January 2019.
56. Ruben Chazarra-Gil, Stijn van Dongen, Vladimir Yu Kiselev, and Martin Hemberg. Flexible comparison of batch correction methods for single-cell RNA-seq using BatchBench. May 2020.
57. Krzysztof Polanski, Matthew D Young, Zhichao Miao, Kerstin B Meyer, Sarah A Teichmann, and Jong-Eun Park. BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics*, 36(3):964–965, February 2020.
58. Korsunsky, I., Millard, N., Fan, J. et al. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat Methods* 16, 1289–1296 (2019).
59. Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length and helmholtz free energy. In J D Cowan, G Tesauro, and J Alspecter, editors, *Advances in Neural Information Processing Systems 6*, pages 3–10. Morgan-Kaufmann, 1994.
60. Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, August 2013.
61. Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. February 2017.
62. Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(Mar):723–773, 2012.
63. Valentine Svensson. Droplet scRNA-seq is not zero-inflated. *Nat. Biotechnol.*, 38(2):147–150, February 2020.

64. Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for Domain-Invariant representation learning. February 2017.
65. Werner Zellinger, Bernhard A Moser, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Robust unsupervised domain adaptation for neural networks via moment alignment. *Inf. Sci.*, 483:174–191, May 2019.
66. Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. HoMM: Higher-order moment matching for unsupervised domain adaptation. December 2019.
67. Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised Multi-Target domain adaptation: An information theoretic approach. October 2018.
68. Rao, A., Barkley, D., França, G.S. *et al.* Exploring tissue architecture using spatial transcriptomics. *Nature* **596**, 211–220 (2021).
69. Methods and systems for determining gene expression profiles and cell identities from multi-omic imaging data. U.S. Patent No. US20220180975A1, 2022.



## **Part A:**

### *Evolution & Evolvability*

# The evolution, evolvability and engineering of gene regulatory DNA

Eeshit Dhaval Vaishnav<sup>1,2,12</sup>✉, Carl G. de Boer<sup>3,4,12</sup>✉, Jennifer Molinet<sup>5,6</sup>, Moran Yassour<sup>4,7,8</sup>, Lin Fan<sup>2</sup>, Xian Adiconis<sup>4,9</sup>, Dawn A. Thompson<sup>2</sup>, Joshua Z. Levin<sup>4,9</sup>, Francisco A. Cubillos<sup>5,6</sup> & Aviv Regev<sup>4,10,11</sup>✉



**Paper:** <https://doi.org/10.1038/s41586-022-04506-6>

**Code:** <https://github.com/1edv/evolution>

**Data:** <https://bit.ly/EvolutionZenodo>

**App:** <https://1edv.github.io/evolution/>

**Nature Cover:** <https://www.nature.com/nature/volumes/603/issues/7901>

**Contribution:** First (and a corresponding) author

# The evolution, evolvability, and engineering of gene regulatory DNA

Eeshit Dhaval Vaishnav<sup>1,2,11§</sup>, Carl G. de Boer<sup>3,8,11§</sup>, Jennifer Molinet<sup>4,5</sup>, Moran Yassour<sup>6,7,8</sup>, Lin Fan<sup>2</sup>, Xian Adiconis<sup>8,9</sup>, Dawn A. Thompson<sup>2</sup>, Joshua Z. Levin<sup>8,9</sup>, Francisco A. Cubillos<sup>4,5</sup>, & Aviv Regev<sup>8,10,12§</sup>

<sup>1</sup>Massachusetts Institute of Technology, Cambridge, MA 02139, USA

<sup>2</sup>Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

<sup>3</sup>School of Biomedical Engineering, University of British Columbia, Vancouver, BC, V6T 1Z3, Canada

<sup>4</sup>Universidad de Santiago de Chile, Facultad de Química y Biología, Departamento de Biología, Santiago, 9170022, Chile.

<sup>5</sup>ANID – Millennium Science Initiative Program - Millennium Institute for Integrative Biology (iBio). Santiago, 7500574, Chile.

<sup>6</sup>Faculty of Medicine, The Hebrew University of Jerusalem, Jerusalem 91121, Israel

<sup>7</sup>The Rachel and Selim Benin School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 91904, Israel

<sup>8</sup>Klarman Cell Observatory, Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

<sup>9</sup>Stanley Center for Psychiatric Research, Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

<sup>10</sup>Howard Hughes Medical Institute and Koch Institute of Integrative Cancer Research, Department of Biology, Massachusetts Institute of Technology, Cambridge, MA 02140 USA

<sup>11</sup>These authors contributed equally

<sup>12</sup>Current address: Genentech, 1 DNA Way, South San Francisco, CA, USA

<sup>§</sup>Correspondence should be addressed to: [edv@mit.edu](mailto:edv@mit.edu), [carl.deboer@ubc.ca](mailto:carl.deboer@ubc.ca), [aviv.regev.sc@gmail.com](mailto:aviv.regev.sc@gmail.com)



## SUMMARY

Mutations in non-coding regulatory DNA sequences can alter gene expression, organismal phenotype, and fitness<sup>1-3</sup>. Constructing complete fitness landscapes, mapping DNA sequences to fitness, is a long-standing goal in biology, but has remained elusive because it is challenging to generalize reliably to vast sequence spaces<sup>4-6,8-17</sup>. Here, we construct sequence-to-expression models that capture fitness landscapes and use them to decipher principles of regulatory evolution<sup>7</sup>. Using millions of randomly-sampled promoter DNA sequences<sup>18</sup> and their measured expression levels in the yeast *Saccharomyces cerevisiae*, we learn deep neural network models that generalize with excellent prediction performance, and enable sequence design for expression engineering<sup>19</sup>. Using our models, we study expression divergence under genetic drift and strong-selection weak-mutation regimes<sup>20-23</sup> to find that regulatory evolution is rapid and subject to diminishing returns epistasis, that conflicting expression objectives in different environments constrain expression adaptation, and that stabilizing selection on gene expression leads to the moderation of regulatory complexity. We present an approach for using our models to detect signatures of selection on expression from natural variation in regulatory sequences and use it to discover an instance of convergent regulatory evolution. We assess mutational robustness, finding that regulatory mutation effect sizes follow a power law, characterize regulatory evolvability, visualize promoter fitness landscapes, discover evolvability archetypes and highlight the mutational robustness of natural regulatory sequence populations<sup>24-25</sup>. Our work provides a general framework for addressing fundamental questions in regulatory evolution.

## RESULTS

### Models predict expression from sequence

We begin by building models that predict gene expression given an 80 bp promoter DNA sequence. To train these models, we measure the expression driven by promoter sequences using an approach we previously described<sup>26</sup>, where 80 bp of DNA are embedded within a promoter construct and the associated expression is assayed in the *S. cerevisiae* (**Methods**). We clone promoter sequences into an episomal low copy number YFP expression vector, transform them into yeast, culture the yeast in the desired media, sort the yeast into 18 expression bins, and sequence the promoters present from the yeast in each bin to estimate expression (**Methods** and **Supplementary Information**). To avoid biases<sup>5</sup> towards extant sequences, we measured the expression of 80 bp random DNA sequences, where each base is randomly sampled from the four bases. For training data, we measured each of >30 million sequences in complex media (YPD, **Methods**) and >20 million sequences in defined media (SD-Ura, synthetic defined lacking uracil). Using the resulting pairs of sequences and measured YFP expression levels, we trained convolutional neural network models (“convolutional models”) that predict expression from sequence in each medium (**Methods**).

To show that the learned convolutional models generalize to new sequences, we predicted the expression for several sets of test sequences not seen during model training, and compared them to their experimentally measured levels (**Methods**). For these test sequences, we quantified expression in independent experiments using the same experimental approach and in the same media. Our convolutional models had excellent prediction performance on native yeast promoter

test sequences (Pearson's  $r = 0.960$ ,  $P < 5 \cdot 10^{-324}$ ,  $n=61,150$ ; **Fig. 1b**), and on multiple other test sets in both complex and defined media (**Extended Data Fig. 1**).

These results represent a ~45% decrease in error compared to the performance of biochemical models we previously<sup>26</sup> trained on the same data (complex media; native yeast promoter test sequences; **Supplementary Notes and Methods**). Other published genomic model architectures adapted to and trained using our data also had excellent performance (**Supplementary Fig. 4a**), highlighting the predictive power of deep neural network models trained using our large-scale data. Finally, the expression measurements were highly correlated for the same sequences between the two media (Pearson's  $r = 0.978$ , **Extended Data Fig. 2a**) and models trained on defined medium predicted expression in complex medium well (Pearson's  $r = 0.966$ , **Extended Data Fig. 2b**). However, for some sequences we expect differences between growth conditions (below).

### **Models enable expression engineering**

We leveraged the high predictive performance of our convolutional models for a synthetic biology application of gene expression engineering, by using model predictions as a 'fitness function' for genetic algorithms (GA) to design sequences with extreme expression values. We initialized the GA with a population of 100,000 randomly-generated samples from the sequence space, and simulated 10 generations to maximize (or minimize) the expression output from the convolutional model (**Methods**). We then synthesized the 500 sequences with the top predicted maximum (or minimum) expression levels and tested them experimentally. The GA-designed sequences drove, on average, more extreme expression than >99% of native sequences (99.6% for high expressing; 99.3% for low), with ~20% of designed sequences yielding more extreme expression than any

native sequence tested (23.5% for high; 18.4% for low) (**Fig. 1c**). Thus, our sequence-to-expression model can be used for gene expression engineering.

### **Expression diverges under genetic drift**

We next assessed the evolutionary malleability of expression under different evolutionary scenarios: random genetic drift, stabilizing selection, and directional selection for extreme expression levels (**Fig. 2**). In each case, we first simulated the scenario, using our convolutional model to predict the expression for each sequence, and then tested the model's evolved sequences experimentally, where possible (**Methods**).

We first simulated random genetic drift of regulatory sequences, with no selection on expression levels. We randomly introduced a single mutation in each random starting sequence, repeated this process for multiple consecutive generations, and used our convolutional model to predict the difference in expression between the mutated sequences in each trajectory relative to the corresponding starting sequence (**Fig. 2a-c**). Expression levels diverged as the number of mutations increased, with 32 mutations in the 80 bp region resulting in nearly as different expression from the original sequence as two unrelated sequences (**Fig. 2b**). We validated our results experimentally by synthesizing sequences with zero to three random mutations and measuring their expression in our assay (**Methods**). The experimental measurements closely matched our predictions in both complex (**Fig. 2c**) and defined (**Extended Data Fig. 1e**) media, both in expression change (Pearson's  $r$ : 0.869 and 0.847, respectively; **Extended Data Fig. 1h,i**) and level (Pearson's  $r$ : 0.973 and 0.963 respectively; **Extended Data Fig. 1l,m**).

## Stabilizing selection tempers complexity

Although gene regulatory networks often appear to be highly interconnected<sup>26,27</sup>, the sources of this regulatory complexity and how it changes with the turnover of regulatory mechanisms<sup>28</sup> remain unclear. We used our model to study the evolution of regulatory complexity in the context of stabilizing selection, which favors the maintenance of existing expression levels. We first quantified regulatory complexity, defined as 1 minus the Gini coefficient (a measure of inequality of continuous values within a population) of TF regulatory interaction strengths. For this, we used an interpretable biochemical model we previously developed<sup>26</sup> (**Methods**) because it has parameters that explicitly correspond to TFs, and we can directly query their contributions to model predictions. Next, starting with native sequences whose regulatory complexity is either extremely high (many TFs with similar contributions to expression) or low (few TFs contribute disproportionately to expression) and spanning a range of expression levels, we introduced single mutations into each starting native sequence for each of 32 consecutive generations, identified the sequences that conserved the original expression level using the convolutional model, and selected one of them at random for the next generation. We then assessed the regulatory complexity of the evolved sequences.

As random mutations accumulated, the regulatory complexity of sequences starting at both complexity extremes shifted towards moderate complexities (**Fig. 2d**, rightmost blue and orange), closer to the averages for both random and native sequences (**Fig. 2d**, greys). This suggests that stabilizing selection on expression leads to a moderation of regulatory complexity, resulting from gradual drift in the roles of the different regulators, such as an increase in complexity due to a decrease in the relative contribution of one predominant TF (*e.g.* Abf1p for *AIF1*), or a decrease

in complexity through smaller changes in a much larger number of sites (*e.g.* *YDR476C*; **Supplementary Fig. 8**). The overall distribution of regulatory complexity of native yeast promoters is similar to that of random sequences (**Fig. 2d**, grey boxes), suggesting that there is little selection on the regulatory complexity of native sequences in a single environment.

### **Strong selection rapidly finds extrema**

To study the impact of directional selection on expression, we simulated the strong-selection weak-mutation (SSWM) regime<sup>29</sup> (**Fig. 2e**, **Methods**), where each mutation is either beneficial or deleterious (strong selection, with mutations surviving drift and fixing in an asexual population), and mutation rates are low enough to only consider single base substitutions during adaptive walks (weak mutation). Starting with a set of native promoter sequences, at each iteration (generation), for a given starting sequence of length  $L$ , we considered all of its  $3L$  single-base mutational neighbors, used our convolutional model to predict their expression, and took the sequence with the largest increase (or separately, decrease) in expression at each iteration (generation) as the starting sequence for the next generation (**Fig. 2e**, **Methods**).

Sequences that started with diverse initial expression levels rapidly evolved to high (or separately, low) expression, with the vast majority evolving close to saturating extreme levels within 3-4 mutations in both the complex (**Fig. 2f**) and defined (**Extended Data Fig. 1f**) media. Sequences took diverse paths to evolve either high or low expression (**Supplementary Fig. 7**). We validated these trajectories experimentally for select series of sequences (**Fig. 2g**, **Extended Data Fig. 1g**), measuring the expression driven by synthesized sequences from several generations along

simulated mutational trajectories for complex media (10,322 sequences from 877 trajectories) and defined media (6,304 sequences from 637 trajectories). We observed extreme expression within 3-4 mutational steps, with high agreement between measured and predicted expression change (**Extended Data Fig. 1j,k**; Pearson's  $r$ : 0.977 and 0.948, respectively) and expression levels (**Extended Data Fig. 1n,o**; Pearson's  $r$ : 0.980 and 0.963) along the trajectories in both complex and defined media. Thus, *cis*-regulatory sequence evolution is rapid and subject to diminishing returns epistasis<sup>30</sup>.

### **Opposing objectives constrain adaptation**

In contrast to the rapid evolution towards expression extremes, we found that evolution to satisfy two opposing expression requirements (one in each growth media) was more constrained. A concrete example is the expression of the *URA3* gene: organismal fitness *increases* with increased *URA3* expression in defined media lacking uracil, because Ura3p is required for uracil biosynthesis, but fitness *decreases* with increased *URA3* expression in complex media containing 5-FOA due to Ura3p-mediated conversion of 5-FOA to toxic 5-fluorouracil (**Extended Data Fig. 2c**). To study this regime<sup>31</sup>, we started with a set of native promoter sequences (and separately, a set of random sequences) and used the convolutional model to simulate SSWM trajectories (**Methods**) that maximize the *difference* in expression between the two media (defined and complex). While the difference in expression increased with each generation (**Extended Data Fig. 2d,e**), the vast majority of sequences achieved neither the maximal nor the minimal expression in either condition after 10 generations (**Fig. 2h**, **Extended Data Fig. 2f**), for both native and random starting sequences. The evolved sequences became enriched for motifs for TFs involved in nutrient sensing and metabolism, compared to the starting sequences (**Extended Data Fig. 2g**), suggesting

that the model is taking advantage of subtle differential activity of certain regulators between the two conditions to evolve condition specificity. Thus, while evolving a sequence to achieve a single expression optimum requires very few mutations, encoding multiple opposing objectives in the same sequence is more difficult, limiting expression adaptation.

### **Transformers enable inference at scale**

We next turned to the evolution and evolvability of regulatory sequences in extant strains and species. This required us to predict expression for billions of sequences and, although our convolutional model had excellent predictive power, our implementation was limited in its scalability and incompatible with the Tensor Processing Units (TPUs), available to us for larger-scale computational tasks (**Methods**). To enable large-scale expression prediction, we developed “transformer” models that used transformer encoders<sup>32</sup> with other building blocks attempting to implicitly capture known aspects of regulation<sup>33</sup> (**Methods, Supplementary Fig. 12**). The transformer models had ~20x fewer parameters than the convolutional models (**Methods, Supplementary Information**), predicted expression as well as the convolutional models (**Extended Data Fig. 3**), and better captured the propensity for expression to plateau under SSWM (**Supplementary Fig. 19**). The convolutional and transformer models had highly correlated predictions in both media (**Supplementary Fig. 4e-h**, Pearson’s  $r=0.967-0.985$ ), and yielded equivalent conclusions from the analyses of genetic drift, directional selection and conflicting objectives (**Extended Data Fig. 3, Supplementary Fig. 17-18**).



## The Expression Conservation Coefficient

We applied our sequence-to-expression transformer model to detect evidence of selective pressures on natural regulatory sequences, inspired by the way in which the ratio of non-synonymous (“non-neutral”) to synonymous (“neutral”) substitutions ( $d_N/d_S$ ) in protein coding sequences is used estimate the strength and mode of natural selection<sup>34</sup>. By analogy<sup>2,35</sup>, for regulatory sequences<sup>2</sup>, we used the transformer model to quantitatively assess the impact of naturally occurring regulatory genetic variation on expression, compared to that expected with random mutations, and summarized this with an Expression Conservation Coefficient (ECC) (**Methods**). To compute the ECC, we compared, for each gene’s promoter, the standard deviation of the expression distribution predicted by the transformer model for a set of naturally varying orthologous promoters ( $\sigma_B$ ) to the standard deviation of the expression distribution predicted for a matched set of random variation introduced to that promoter ( $\sigma_C$ ; related to the mutational variance<sup>36</sup>; **Fig. 3a**). We define the ECC for a gene as  $\log(\sigma_C/\sigma_B)$ , such that a positive ECC indicates stabilizing selection on expression (lower variance in native sequences than expected by chance), a negative ECC indicates diversifying (disruptive) selection or local adaptation (greater variance in native sequences), and values near 0 suggest neutral drift.

We calculated the ECC for 5,569 *S. cerevisiae* genes using the natural variation observed across over 4.73 million orthologous promoter sequences from the 1,011 *S. cerevisiae* isolates<sup>37</sup> in the -160 to -80 regions (with respect to the Transcription Start Site (TSS)), a critical location for TF binding<sup>38</sup> and determinant of promoter activity<sup>26</sup> (**Fig. 3a,b, Supplementary Table 1**), using our transformer model to predict the expression for each sequence. To assess the robustness of the ECC values, we recomputed the ECC using multiple published sequence-to-expression model

architectures that we adapted and trained using our data and found that models with similarly high predictive power resulted in similar ECC values (**Supplementary Fig. 4b-d, 5g**).

Over 70% of promoters had positive ECCs, suggesting stabilizing selection (and conserved expression) (binomial test  $P < 10^{-215}$ ) (**Fig. 3b**), consistent with previous reports based on direct measurements of gene expression<sup>39</sup>. Genes with high ECCs were enriched in highly-conserved core cellular processes (*e.g.*, RNA and protein metabolism) (**Fig. 3b, Supplementary Table 2**), and those with low ECCs were most enriched in processes related to carboxylic acid and alcohol metabolism (**Fig. 3b, Supplementary Table 2**), potentially reflecting adaptation of fermentation genes to the diverse environments of these isolates<sup>37</sup>.

### **The ECC discovers convergent evolution**

A striking example of predicted positive selection is the promoter of *CDC36* (*NOT2*; ECC= -2.138, **Fig. 3b**), which has common natural alleles with either low or high (predicted) expression across the isolates (**Fig. 3c**). Analysis of *CDC36* promoter sequences (**Methods**) suggests that low-expression evolved at least twice independently, resulting in two distinct variants with reduced expression (**Fig. 3c**, allele 1 and 2). Interrogation with the biochemical model<sup>26</sup> to identify factors impacting these expression differences (**Extended Data Fig. 4a**) suggested that both low-expression alleles are explained by disruption of the same binding site for Upc2p, an ergosterol sensing TF (**Fig. 3c**). To validate this, we restored the putative Upc2p binding site in a strain (WE), where it is otherwise disrupted, and measured expression levels by qPCR and growth upon changing carbon source (**Methods**). Restoration of the Upc2p binding site increased actual

expression, confirming the model's prediction (Pearson's  $r=0.96$ ,  $p=0.039$ ,  $n=4$ ; **Fig. 3d**). We hypothesized that these variants could alter the rate of transcriptional reprogramming when changing environments via Cdc36p-regulated mRNA turnover<sup>40</sup>. Indeed, restoration of the Upc2 binding site reduced the strains' lag time to growth when switching carbon sources (**Fig. 3d**, right; **Methods**), and they grew to a higher culture density (**Supplementary Fig. 10**). Thus, convergent evolution of the *CDC36* promoter, discovered using the ECC, independently produced two alleles that result in similar perturbations to TF binding, expression, and growth.

### **ECC vs. cross-species RNAseq and fitness**

ECC values were consistent with expression conservation as measured for yeast orthologs across clades at short (*Saccharomyces*), medium (Ascomycota), or long (mammals) evolutionary scales (**Extended Data Fig. 4b**). In *Saccharomyces*, 1:1 orthologs with conserved expression levels across species (as measured by RNA-seq<sup>41</sup>) had significantly higher ECC (computed from the 1,011 yeast isolates) than genes whose expression was not conserved (two-sided Wilcoxon rank-sum  $P = 3.1 \times 10^{-4}$ , **Extended Data Fig. 4b, bottom left, Methods**). Next, we performed RNA-seq across 11 Ascomycota yeast species (**Methods**), finding that 1:1 orthologs with conserved expression across Ascomycota had significantly higher ECC values (**Extended Data Fig. 4b, bottom center**,  $P = 1.16 \times 10^{-6}$ ). Finally, the 1:1 orthologs of genes with high ECC values in the 1,011 *S. cerevisiae* isolates also had more conserved expression within mammals<sup>42</sup> (**Extended Data Fig. 4b, bottom right**,  $P = 1.07 \times 10^{-4}$ , **Methods**). Thus, while 1:1 yeast-mammal orthologs are likely critical to an organism's fitness, only a subset of these may be under stabilizing selection on expression, and this subset tends to be under such selection in both yeasts and mammals. Thus,

the ECC quantifies stabilizing selection on expression in yeast and may predict stabilizing selection on orthologs' expression in other species.

Genes with higher ECCs also had a stronger effect on fitness in *S. cerevisiae* upon changing their expression level. We interrogated the total variation of previously measured expression-to-fitness curves<sup>11</sup> to calculate a 'fitness responsivity' score that captures the dependence of fitness on expression (**Extended Data Fig. 5, Methods**). Fitness responsivity was significantly positively correlated with the ECC (**Supplementary Fig. 2e**,  $P = 0.003$ , Spearman  $\rho = 0.326$ ). Fitness responsivity was not associated with regulatory sequence divergence *per se* across the promoter sequence (as estimated by mean Hamming distance among orthologous promoters, **Methods, Supplementary Fig. 2d**,  $P = 0.46$ , Spearman  $\rho = 0.083$ ). Thus, while stabilizing selection on gene expression (as captured by the ECC) can shape the types of mutations that accumulate in the population, it may have little effect on the overall rate at which mutations accumulate in promoter regions within populations, which has been previously used to test for evidence of selection.

### **Stabilizing selection shapes robustness**

While a gene's ECC (computed from the natural genetic variation in regulatory DNA) represents the imprint of its evolutionary history, its mutational robustness (assessed directly from the gene's promoter sequence) should describe how *future* mutations would affect its expression<sup>43</sup>. Across all native yeast promoters, the magnitude of expression changes predicted by the transformer model due to single base-pair mutations follows a power law with an exponent of 2.252 (standard error of fit  $\sigma = \pm 0.002$ ,  $P = 2.4 \times 10^{-263}$ ), such that a small number of mutations have an outsized effect

on expression (~10% of mutations account for ~50% of the changes in expression, **Extended Data Fig. 4d**). In individual genes, the distribution can vary substantially (below).

For a given promoter sequence, we defined the mutational robustness of a sequence length  $L$ , as the percent of its  $3L$  single nucleotide mutational neighbors predicted by the transformer model to result in a *negligible* change in expression (**Extended Data Fig. 4c, Methods**), following previous definitions of mutational robustness<sup>25,43</sup>. The mutational robustness of a gene's promoter sequence was positively correlated with the gene's fitness responsivity (**Supplementary Fig. 2f**, Spearman  $\rho = 0.476$ ,  $P = 8.18 \times 10^{-6}$ ), suggesting that fitness-responsive genes have evolved more mutationally robust regulatory sequences. Mutational robustness, which, unlike the ECC, is computed for single sequences without a set of variants across a population, was also correlated to the ECC (**Supplementary Fig. 2g**, Spearman  $\rho = 0.515$ ,  $P = 9.99 \times 10^{-7}$ ). Similarly, the promoter sequences of yeast genes with conserved expression across *Saccharomyces* strains<sup>41</sup>, Ascomycota species, or mammals<sup>42</sup> had higher mutational robustness ( $P = 8.4 \times 10^{-3}$ ,  $6.5 \times 10^{-5}$ , and 0.00377, respectively, two-sided Wilcoxon rank-sum test).

Thus, genes whose expression levels are under stabilizing selection have regulatory sequences that tend to be more robust to the impact of mutations, which may reflect their history and constrain their future.

### **Fitness landscapes in evolvability space**

Mutational robustness enables the exploration of novel genotypes that could subsequently facilitate adaptation and thus promote evolvability, the ability of a system to generate heritable phenotypic

variation<sup>25</sup>. To characterize regulatory evolvability, we extended our description of mutational robustness by representing each sequence using a sorted vector of expression changes (predicted by the transformer model) that are accessible through single nucleotide mutations (**Fig. 4a**, left, **Methods**). This ‘evolvability vector’ captures the capacity for changes in genotype to alter expression phenotype, in line with previous definitions of evolvability<sup>25</sup>.

We next asked whether regulatory evolvability vectors fell into distinct classes by identifying evolvability ‘archetypes’. Archetypes<sup>44</sup> represent the extremes of canonical patterns, such that the evolvability vector of each individual sequence can be represented by its similarity to each of several archetypes representing these extremes. Applying this paradigm, we used our transformer model to compute evolvability vectors for a new random sample of a million sequences and then learned a two-dimensional representation of these evolvability vectors (referred to as the ‘evolvability space’) using an autoencoder<sup>45</sup> (**Fig. 4a**, right, **Methods**). This archetypal evolvability space, that is bounded by a simplex whose vertices represent evolvability archetypes (**Fig. 4a**, right, **Methods**) and where the evolvability vector of each sequence is a single point, allows us to effectively visualize arbitrarily large sequence spaces in two dimensions.

Three archetypes captured most of the variation in evolvability vectors (**Extended Data Fig. 6a,b**; **Methods**), corresponding to local expression minimum ( $A_{\text{minima}}$ ), local expression maximum ( $A_{\text{maxima}}$ ), and malleable expression ( $A_{\text{malleable}}$ ) (**Fig. 4b**).  $A_{\text{minima}}$  and  $A_{\text{maxima}}$  correspond to sequences where most  $3L$  mutational neighbors do not change expression, and the ones that do, increase it (for  $A_{\text{minima}}$ ) or decrease it (for  $A_{\text{maxima}}$ ). Conversely, for  $A_{\text{malleable}}$  sequences, most  $3L$  mutational neighbors change expression and are equally likely to decrease or increase it (**Fig. 4b**).

In addition to these three archetypes, mutationally robust sequences were present as a central cleft in the evolvability space (**Fig. 4b,c**; “robust”). The evolvability space also distinguishes native regulatory sequences by their associated expression level (**Fig. 4d**), with intermediate expression more likely to be near the malleable archetype ( $A_{\text{malleable}}$ ) and depleted near the robustness cleft (**Fig. 4d, Supplementary Information**).

The location of sequences in evolvability space reflects the selective pressures operating on the sequence. Sequences under strong stabilizing selection on gene expression tend to be located far away from the malleable archetype: there is a strong negative correlation between malleable archetype proximity and mutational robustness (**Extended Data Fig. 6c,e**; Spearman's  $\rho = -0.746$ ,  $P = 1.97 \times 10^{-15}$ ), the ECC (**Extended Data Fig. 6d,f,g**;  $\rho = -0.596$ ,  $P = 5.4 \times 10^{-9}$ ), fitness responsivity (**Extended Data Fig. 6h**;  $\rho = -0.413$ ,  $P = 1.4 \times 10^{-4}$ ), and expression conservation across species as measured by RNA-seq (*Saccharomyces*:  $P = 0.000251$ , Ascomycota:  $P = 0.00002$ , Mammals:  $P = 0.00114$ ; two-sided Wilcoxon rank-sum test).

To visualize promoter fitness landscapes in two dimensions we combined our sequence-to-expression transformer model with previously measured expression-to-fitness curves<sup>11</sup>, and integrated them with the two-dimensional archetypal evolvability space (**Fig. 4e, Extended Data Fig. 7, Methods**). Unlike prior visualizations of fitness landscapes, which group sequences by their sequence similarity, here, sequences are arranged by the similarity in their evolvability. This approach effectively visualizes arbitrarily large sequence spaces in two-dimensions, as well as groups sequences by their evolutionary properties. This addresses the challenges otherwise posed by sequence similarity-based landscapes since highly similar regulatory sequences can have

different functional properties (*e.g.*, due to a loss of a TF binding site), while very different sequences can be functionally similar (*e.g.*, due to shared TF binding sites). When organismal fitness is available for a particular gene and overlaid on the landscape (**Fig. 4e**, **Extended Data Fig. 7**), the resulting patterns depend on both the condition-specific sequence-to-expression function (*e.g.*, governing color (fitness) through predicted expression, and embedded position, through evolvability) and the gene- and condition-specific expression-to-fitness functions.

Finally, we studied how natural yeast sequences explored evolutionary space, by placing the evolvability vectors of each of set of orthologous promoters of the 1,011 sequenced *S. cerevisiae* isolates<sup>37</sup> in the archetypal evolvability space. When a gene's promoter from one strain is near the malleable archetype, its orthologs in the other strains tended to broadly distribute in the evolvability space (**Extended Data Fig. 6i**), but avoid the robustness cleft (*e.g.*, the *DBP7* promoter from strain S288C; **Fig. 4f**). Conversely, when a promoter is near the robustness cleft (*e.g.*, the *UTH1* promoter from S288C), so are its orthologs (**Fig. 4g**, **Extended Data Fig. 6i**). Using *in silico* mutagenesis to interpret our model, we found that the *DBP7* promoter is particularly malleable partly as a result of an intermediate affinity Rap1p binding site, where the most impactful mutations increased or decreased the Rap1p affinity for this site, impacting expression (**Extended Data Fig. 8a**). By contrast, the *UTH1* promoter requires many sequential mutations, each of which has minimal impact individually, to reduce expression appreciably (**Extended Data Fig. 8b**). This could reflect the ways in which stabilizing selection constrains evolvability: promoters that are not under strong stabilizing selection explore expression space more freely and can quickly adapt to a new expression optimum, since the population likely already contains multiple alleles that achieve



diverse expression levels (*e.g.* **Fig. 4f**). Interestingly, many of the native sequences in *S. cerevisiae* are near the robustness cleft (**Fig. 4h**).

Thus, the evolvability vector, which can be computed using our model directly for any sequence (without any population genetics data), encodes information about a sequence's evolutionary history and potential futures.

## DISCUSSION

Here, we presented a framework that addresses fundamental questions in the evolution and evolvability of regulatory sequences<sup>2,25</sup>. Our models, developed using a combination of large scale random sequence libraries, sensitive reporter assays and deep learning (**Methods**), are useful as “oracles” for model-guided biological sequence design<sup>19</sup>, and answering important questions in the study of fitness landscapes<sup>4-6</sup>, evolutionary malleability of expression and its variation across strains and species<sup>2</sup>, mutational robustness<sup>43</sup>, and evolvability<sup>25</sup>. The framework presented here will help advance synthetic biology, cell and gene therapy, and metabolic engineering in addition to the study of evolution.

Previous studies suggested that evolution favors more complex regulatory solutions<sup>46</sup>, but we showed that if stabilizing selection acts only on expression, regulatory complexity extremes gradually move towards the moderate complexity levels observed in native and random sequences (**Fig. 2d**). This supports a model where most extant regulatory sequences evolved by sampling constraint-satisfying solutions in proportion to their frequency in the sequence space, without specific consideration of the solution's complexity.

In our study, evolving condition-specificity in a promoter sequence was much slower than simply modifying the expression level. Some yeast genes achieve condition-specificity by including multiple binding sites for condition-responsive TFs. For instance, the *GALI-10* Upstream Activating Sequence contains multiple binding sites for the galactose-responsive Gal4, which are conserved across millions of years, suggesting an ancient origin<sup>47</sup>. Because the size of the regulatory region restricts the number of TF binding site locations, including more TFs and more regulatory sequences per gene (*e.g.* enhancers) may be required for more complex regulatory programs observed in higher eukaryotes<sup>48</sup>.

The  $d_N/d_S$  ratio has been used extensively to characterize the evolutionary rates of protein coding genes<sup>34</sup>, and we developed an analogous<sup>2,35</sup> coefficient, the ECC, for detecting evidence of selection on expression from natural variation across multiple orthologous regulatory sequences in strains of one species. The ECC complements and extends existing measures of expression conservation, since it integrates across the regulatory sequence and is not limited to specific TFs or binding motifs, does not require additional experiments to test the functions of mutations for each regulatory region, and does not rely on detecting non-uniformity in mutation distributions.

Complementing the ECC, mutational robustness as calculated with our model is predictive of selective pressures on individual sequences (**Supplementary Fig. 2f-g**). While we find that strong constraint on the function of regulatory sequences can shape them to be robust to future mutations, it is unlikely that robustness itself is the selected trait, since increased robustness to future mutations is likely to be of little marginal benefit<sup>43</sup>. Instead, this may reflect a secondary benefit

of having evolved decreased expression noise<sup>49,50</sup>, or another as-yet-unknown mechanism. It may also reflect the fact that the sequences of some ancestral promoters may be similar to the mutational neighbors of extant sequences, and, if selective constraints on expression have remained stable, these ancestral and extant sequences likely have similar expression levels.

Based on our model-derived evolvability vectors, sequences spanned an evolvability spectrum from robust to malleable (**Fig. 4c-d,f-h**), and for native regulatory sequences, the magnitudes of accessible mutation effects follows a power law. Evolvability vectors also help visualize fitness landscapes<sup>4</sup> (**Fig. 4e, Extended Data Fig. 7**) and future work can further improve our understanding of their topography<sup>4,5</sup>.

Our sequence-to-expression models are currently limited by regulatory region and species. For example, sequence mutations that affect other regulatory mechanisms (*e.g.*, genomic context, mRNA processing and degradation, regulation by RNA-binding proteins, translational efficiency) can compensate for those that affect transcription. While our models emulated the biological process of our experimental system, as demonstrated by their excellent predictive power, future interpretability studies will shed further light on molecular mechanisms. Finally, for multicellular organisms, selection acts simultaneously on expression levels in many different cell types and environments. As models of gene regulation are created for other species, environments, and regulatory regions, our framework will help provide further insights into regulatory evolution.

## **Acknowledgements**

We thank Google TPU Research Cloud for TPU access, Leslie Gaffney for help with figure preparation, Broad Genomics Platform for sequencing work, Jan-Christian Hütter for advice on fitness responsivity, Jenna Pfiffner-Borges for help with RNA-seq, Ruby Yu, Byron Lee and Nima Jaberri for manuscript feedback and members of the Regev lab for discussions. E.D.V. was supported by the MIT Presidential Fellowship. C.G.D. was supported by a Canadian Institutes for Health Research Fellowship and the NIH (K99-HG009920-01). F.A.C. and J.M. were supported by ANID - Programa Iniciativa Científica Milenio - ICN17\_022. Work was supported by the Klarman Cell Observatory and HHMI. A.R. was an Investigator of the HHMI.

## **Conflict of Interest statement**

A.R. is a co-founder and equity holder of Celsius Therapeutics, an equity holder in Immunitas, and until July 31, 2020 was an S.A.B. member of ThermoFisher Scientific, Syros Pharmaceuticals, Neogene Therapeutics and Asimov. From August 1, 2020, A.R. is an employee of Genentech. The other authors declare no competing interests.

## **Data Availability**

Data generated for this study are available on NCBI's GEO, accession numbers GSE163045 and GSE163866. All models and processed data are available on Zenodo at <https://zenodo.org/record/4436477>.

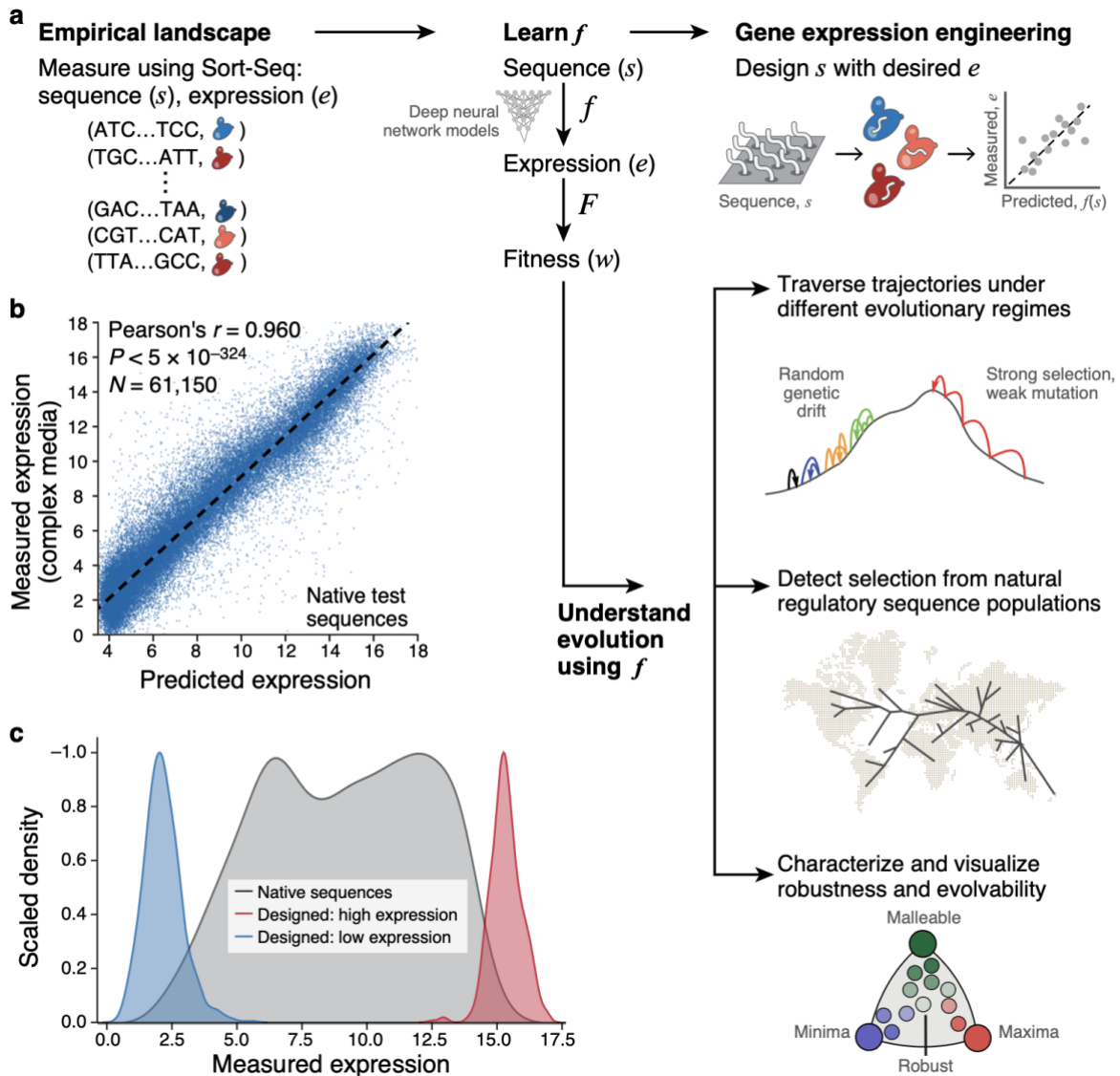
## **Code Availability**

Code is available on GitHub at <https://github.com/ledv/evolution> and CodeOcean at <https://codeocean.com/capsule/8020974/tree>. A web app is available at <https://ledv.github.io/evolution/>.

### **Author contributions**

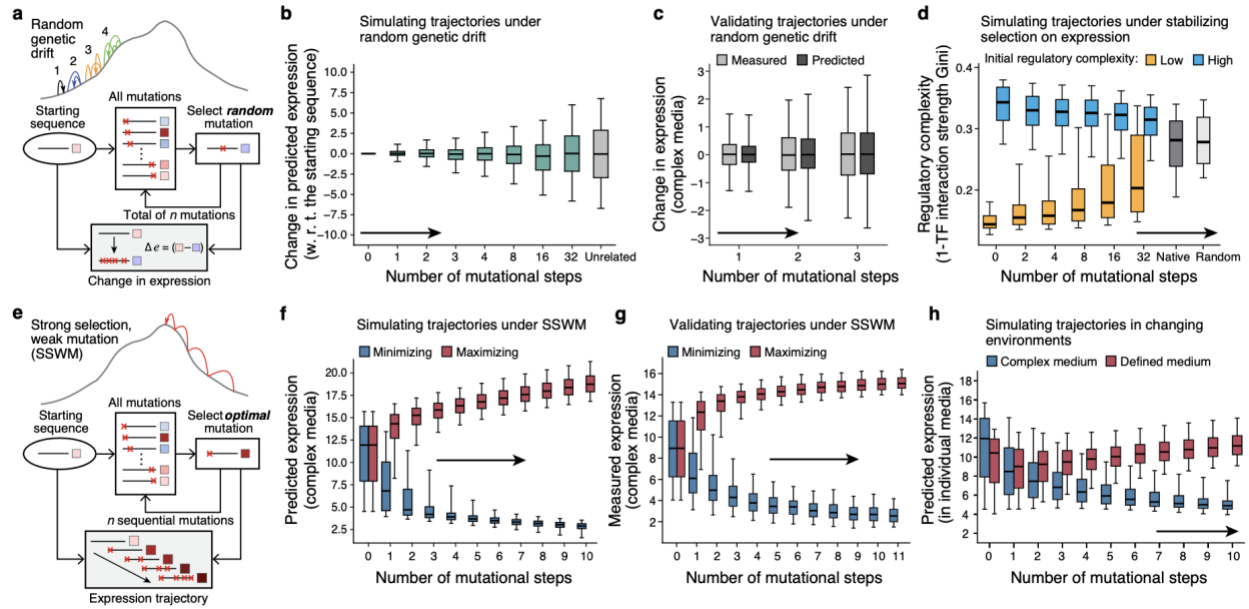
E.D.V., C.G.D. and A.R. conceived, designed and supervised the study. E.D.V. and C.G.D. carried out the analyses. M.Y., L.F., X.A. and D.A.T. performed and J.Z.L. supervised the Ascomycota cross-species RNA-seq experiments. J.M. performed and F.A.C. supervised the *CDC36* experiments. E.D.V. and C.G.D. performed the rest of the experiments. E.D.V., C.G.D. and A.R. wrote the manuscript.

## FIGURES



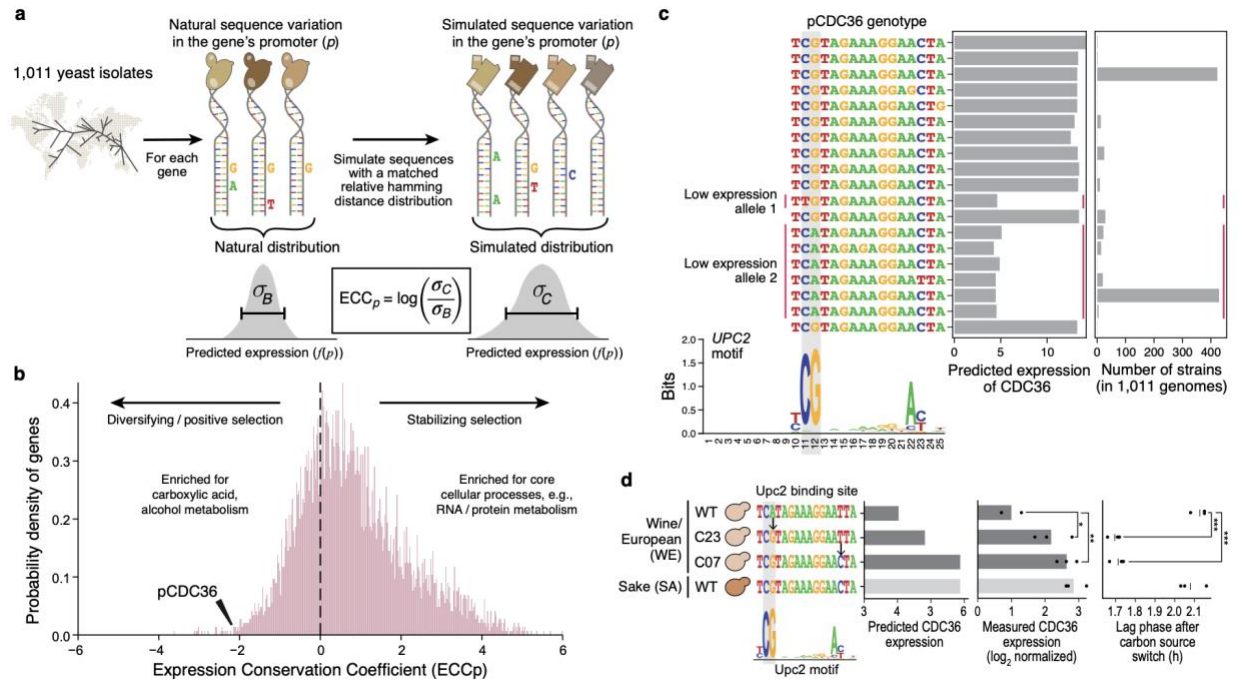
**Fig. 1 | The evolution, evolvability, and engineering of gene regulatory DNA.**

**a**, Project overview. **b**, Prediction of expression from sequence using the model. **b**, Predicted ( $x$  axis) and experimentally measured ( $y$  axis) expression in complex media (YPD) for native yeast promoter sequences. Pearson's  $r$  and associated two-tailed  $p$ -values are shown; dashed line: line of best fit. **c**, Engineering extreme expression values beyond the range of native sequences using a genetic algorithm (GA) and the sequence-to-expression model. Normalized kernel density estimates of the distributions of measured expression levels for native yeast promoter sequences (grey), and sequences designed (by the GA) to have high (red) or low (blue) expression.



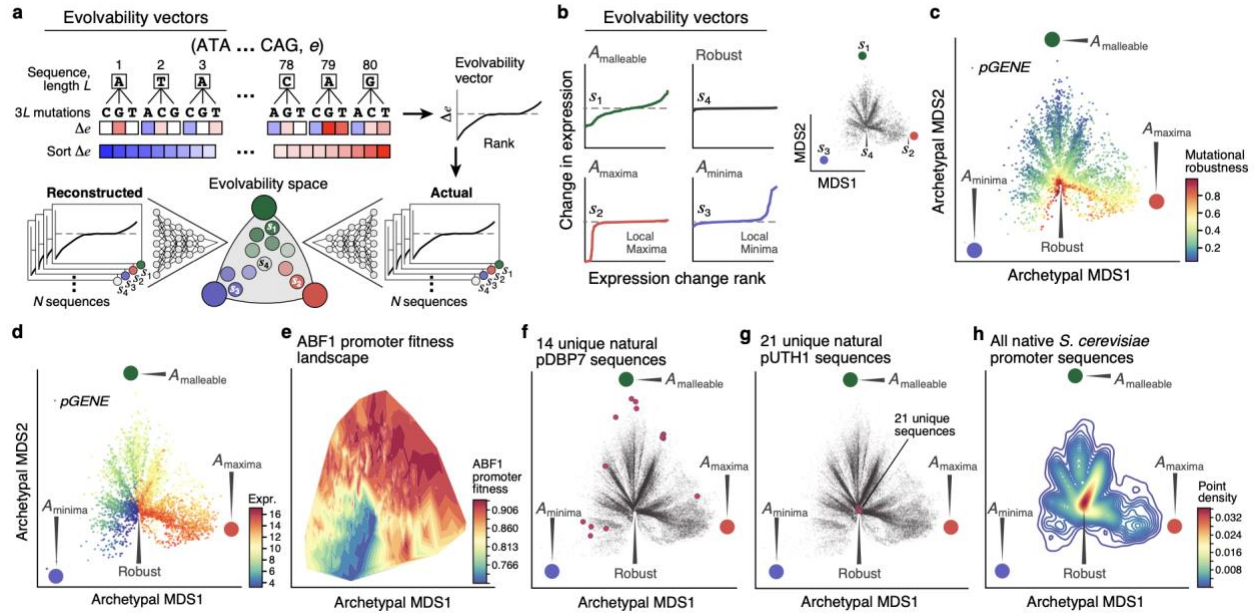
**Fig. 2 | The evolutionary malleability of gene expression.**

**a-c**, Expression divergence under genetic drift. **a**, Simulation procedure. **b**, Predicted expression divergence. Distribution of the change in predicted expression (y axis) for random starting sequences ( $n=5,720$ ) at each mutational step (x axis) for simulated trajectories. Silver bar: expression differences between unrelated sequences. **c**, Experimental validation. Distribution of measured (light grey) and predicted (dark grey) changes in expression in complex media (y axis) for synthesized randomly-designed sequences ( $n=2,983$ ) at each mutational step (x axis). **d**, Stabilizing selection on gene expression leads to moderation of regulatory complexity extremes. Regulatory complexity (y axis) of sequences from sequential mutational steps (x axis) under stabilizing selection to maintain the starting expression levels, where the regulatory interactions of starting sequences are complex (blue;  $n=192$ ) or simple (orange,  $n=172$ ). Right bars: regulatory complexity for native (dark grey) and random (light grey) sequences. **e-g**, Sequences under strong-selection weak-mutation (SSWM) can rapidly evolve to expression optima. **e**, Simulation procedure. **f**, Predicted expression evolution. Distribution of predicted expression levels (y axis) in complex media at each mutational step (x axis) for trajectories favoring high (red) or low (blue) expression, starting with native promoter sequences ( $n=5,720$ ). **g**, Experimental validation. Measured expression distribution in complex media (y axis) for the synthesized sequences ( $n=10,322$  sequences; 877 trajectories) at each mutational step (x axis), favoring high (red) or low (blue) expression. Axis scales differ due to variation in measurement procedure (**Supplementary Information**). **h**, Competing expression objectives constrain expression adaptation. Distribution of predicted expression (y axis) in complex (blue) and defined (red) media at each mutational step (x axis) for a starting set of native promoter sequences ( $n=5,720$ ) optimizing for high expression in defined (red) and simultaneous low expression in complex (blue) media. (**b-d,f-h**) Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range.



**Fig. 3 | The Expression Conservation Coefficient (ECC) detects signatures of stabilizing selection on gene expression using natural genetic variation in regulatory DNA. a,** ECC calculation from 1,011 *S. cerevisiae* genomes<sup>37</sup>. **b,** ECC distribution for *S. cerevisiae* genes. Frequency distribution of ECC values ( $x$  axis). Dashed line separates regions corresponding to disruptive/positive selection (left) and stabilizing selection (right). GO terms enriched by the ECC ranking are shown. Arrowhead: ECC value for the *CDC36* promoter sequence. **c,** Convergent regulatory evolution in the *CDC36* promoter. Predicted expression ( $x$  axis, left bar plot) and associated number of strains ( $x$  axis, right bar plot) of all alleles among the analyzed *CDC36* promoter sequence within 1,011 yeast isolates, along with an alignment of their Upc2p binding site sequences (left; Upc2p binding motif below). Red vertical lines: two independently evolved low-expressing alleles. Grey vertical boxes: key positions in the Upc2p motif with single nucleotide polymorphisms. **d,** Validation of *CDC36* promoter allele expression and organismal phenotype. Strains ( $y$  axis) with different Upc2p binding site alleles for both model-predicted *CDC36* expression (left; predicted on -170:-90 region to capture entire Upc2p binding site), measured *CDC36* expression (middle), and lag phase duration (right). Points: biological replicates ( $n=3$ ); bars/vertical lines: means. Bar color: strain background. Student's t-test p-values, unpaired, equal variance, one-sided (expression, WE WT vs. C23  $p=0.044$ , C07  $p=6.69 \times 10^{-3}$ ) or two-sided (lag phase, WE WT vs. C23  $p=1.34 \times 10^{-4}$ , C07  $p=2 \times 10^{-4}$ ); \* $p<0.05$ ; \*\* $p<0.01$ ; \*\*\* $p<0.001$ .





**Fig. 4 | The evolvability vector captures fitness landscapes.**

**a**, Characterizing regulatory evolvability by computing an evolvability vector. Left and middle: Generating evolvability vectors for a sequence. Right: training an autoencoder with evolvability vectors to generate a 2D representation to visualize sequences in archetypal evolvability space.

**b**, Evolvability archetypes discovered by the autoencoder. Left: Evolvability vectors of the rank ordered ( $x$  axis) predicted change in expression ( $y$  axis) for native sequences closest to each of the malleable (green), maxima (red) or minima (blue) archetypes and the ‘robustness cleft’ (black). Right: all native yeast (*S. cerevisiae* S288C) promoter sequences (grey points) projected onto the archetypal evolvability space by their evolvability vectors. Evolvability archetypes (colored circles) and their closest native sequences ( $s_1$ - $s_4$  as on left) are marked.

**c,d**, Evolvability space captures mutational robustness and expression levels. Evolvability vectors (points) of all native yeast promoter sequences projected onto the evolvability space (archetypes are large colored circles, as in **b**) and colored by mutational robustness (**c**) or predicted expression levels (**d**).

**e**, *ABF1* promoter fitness landscape. Evolvability vectors of promoter sequences projected onto the evolvability space and colored by computed fitness (color, **Methods**).

**f,g**, Malleable promoter sequences dynamically traverse the evolvability space. Evolvability vector projections of native sequences (points) from all 1,011 *S. cerevisiae* isolates. Red points: natural promoter sequence variants for *DBP7*, the promoter closest to the malleable archetype (**f**) and for *UTH1*, the promoter closest to the robustness cleft (**g**).

**h**, The robustness of native promoter sequences. Density (color) of all native yeast promoter sequences when their evolvability vectors are projected onto the evolvability space.

## METHODS

### Experimental measurement of sequence-expression pairs using a Sort-seq strategy

We experimentally measured expression using a Sort-seq<sup>2,3,51-59</sup> strategy called the Gigantic Parallel Reporter Assay (GPRA) we previously described<sup>26</sup> (**Supplementary Fig. 1**). Briefly, for each set of expression measurements mentioned, random or designed single stranded oligonucleotides were ordered from IDT (random; **Supplementary Table 3**) or Twist Biosciences (designed; sequences on GEO; accession GSE163045), cloned into the promoter of a Yellow Fluorescent Protein (YFP) gene within a CEN plasmid (Addgene: 127546) as previously described<sup>26</sup> and transformed into yeast (strain Y8205 for the training dataset of random sequences, and strain S288C::*ura3* for all the rest of the sequences measured). The library is maintained in yeast as an episomal low copy number plasmid. It was previously reported that the expression measurements are highly correlated with expression levels as measured using integrated reporters ( $R^2=0.97$ )<sup>54</sup>. Yeast were grown in continuous log phase, diluting as necessary to maintain an OD between 0.05 and 0.6 for 8-10 generations up until the time of harvest. Cells were harvested, washed once in ice cold PBS, and kept on ice in PBS until sorting. Cells were sorted into 18 uniformly-sized expression bins covering the majority of the expression distribution. Post sort, cells were re-grown in SD-Ura until saturation, plasmids isolated, and sequencing libraries created sequenced with a 150 cycle NextSeq kit. For libraries with random 80 bp sequences, sequences were consolidated as previously described<sup>26</sup>. Reads from other (defined, non-random) libraries were aligned to the pre-defined sequences using Bowtie2<sup>60</sup>, including only reads that perfectly matched a designed sequence. For each sequence, the expression level was the average of the expression bins in which it was observed, weighted by the number of times it was observed in each bin. These expression measurements were carried out separately in defined media lacking uracil

(SD-Ura (Sunrise Science, #1703-500)) and complex media (YPD: yeast extract, peptone, dextrose).

### **Architecture of the convolutional model**

A deep neural network model<sup>20,21,23,61-69</sup> with convolutional layers was constructed and used for designing sequences with high and low expression (**Fig. 1c**), and running evolutionary simulations under stabilizing selection, genetic drift, and SSWM (**Fig. 2**) for each condition. These designed sequences, whose expression was experimentally quantified (e.g. **Fig. 1c** and **2d,g**), were designed using models with the following architecture:

**Input.** The input is the DNA sequence ( $s$ ) represented in one-hot encoding. Input Shape: (1, 110, 4)

#### ***Convolution Block***

- For the forward and reverse strand, separately,
  - o Strand-specific convolution layer 1. Kernel Shape: (1, 30, 4, 256)
  - o Strand-specific convolution layer 2. Kernel Shape: (30, 1, 256, 256)
- Concatenation of features from the forward and reverse strand
- Convolution layer 3. Kernel Shape: (30, 1, 512, 256)
- Convolution layer 4. Kernel Shape: (30, 1, 256, 256)
- A bias term and a ReLU activation was added to each convolution layer in this block.

#### ***Fully Connected Layers***

- Fully connected layer 1. Kernel Shape: (110\*256, 256).

- Fully connected layer 2. Kernel Shape: (256, 256)
- A bias term and a ReLU activation were added to each layer in this block.

**Output.** Linear combination of the 256 features extracted as a result of all the previous operations on the sequence ( $s$ ) to generate the predicted expression ( $e$ ).

Every fully connected layer was  $L2$  regularized with a 0.0001 weight and had a dropout probability of 0.2.

### **Training of the convolutional model**

For training, 20,616,659 random sequences for the defined medium and 30,722,376 random sequences for the complex medium (each to train a separate model) were used, along with their experimentally measured expression as described above. A mini-batch size of 1,024 was used for training and a mean squared error loss was optimized using the Adam optimizer<sup>70</sup> with an initial learning rate of 0.0005. The model was trained for 5 epochs. Model architecture was written in TensorFlow<sup>71</sup> 1.14 using Python 3.6.7. The convolutional model used TensorFlow graphs and sessions in its implementation and was thus incompatible with the Tensor Processing Units (TPUs)<sup>72</sup>. These convolutional models (for both media) were used for all the predictions in **Fig. 1** and **2** and **Extended Data Fig. 1, 2**.

The models were tested by predicting expression on sequences that the model had never seen before (**Supplementary Fig. 21**) that were measured in separate experiments, where the library was lower complexity (fewer sequences) than the experiments that generated the training data,

such that the expression associated with each sequence was measured with high accuracy (~100 yeast cells per sequence on average). The test libraries included random, native (*i.e.* present in the yeast genome), and designed sequences.

Training and evaluation were carried out on 4 Tesla M60 GPUs. All code for training and using the convolutional model is available here:

[https://github.com/ledv/evolution/tree/master/manuscript\\_code/model/gpu\\_only\\_model](https://github.com/ledv/evolution/tree/master/manuscript_code/model/gpu_only_model).

### **Architecture of the transformer model**

A transformer model<sup>23,32,73</sup> was developed to run inference faster than the convolutional model, as needed for the evolutionary analyses in **Fig. 3 and 4**. The transformer model had ~20x fewer parameters (~1.3 million, compared to the ~24 million parameters of the convolutional model) and was able to leverage Tensor Processing Units (TPUs) for computation. Transformer models are used in all the analyses in **Fig. 3 and 4** and **Extended Data Fig. 3-4, and 6-8**. Benchmarking analyses and ablation analyses for the transformer model are available in the **Supplementary Information**.

The deep transformer model has the following architecture (**Supplementary Fig. 12**):

**Input.** The input is the DNA sequence ( $s$ ) represented in one-hot encoding. Input Shape: (110, 4)

**Convolution Block.** The convolution block is constructed in the following order (**Supplementary Fig. 12b**):

- Reverse Complement Aware 1D Convolution. The forward and reverse strand are operated on separately with a convolutional kernel to generate strand specific sequence-environment interaction features. Kernel Shape: (30, 4, 256).
- Batch Normalization
- Rectified Linear Unit (ReLU)
- Concatenation of Features from the forward and reverse strand
- 2D Convolution: Convolve over the combined features from both the strands to capture interactions between strands. Kernel Shape: (2, 30, 4, 256)
- Batch Normalization
- ReLU
- 1D Convolution. Kernel Shape: (30, 64, 64)
- Batch Normalization
- ReLU

**Transformer Encoder Blocks.** Two transformer encoder blocks<sup>32,74,75</sup> are constructed in the following order (**Supplementary Fig. 12c**):

- Multi-Head Attention: 8 heads, capturing relations between features from different positions of (*s*) to compute a representation for the features extracted from the convolution block from (*s*).
- Residual Connection
- Layer Normalization
- Feed Forward Layer with 8 units
- Residual connection
- Layer Normalization

***Bidirectional LSTM layer.*** A bidirectional LSTM layer to capture the long-range interactions between different regions of the sequence with 8 units and 0.05 dropout probability.

***Fully Connected Layers (Supplementary Fig. 12d).*** Two Fully connected layers with 64 Hidden Units, each consisting of ReLU and Dropout (0.05 dropout probability).

***Output.*** Linear Combination of 64 features extracted as a result of all the previous operations on the sequence ( $s$ ) to generate the predicted expression ( $e$ ).

### **Training of the transformer model**

20,616,659 random sequences (defined medium) and 30,722,376 random sequences (complex medium), along with their experimentally measured expression, were used to train separate models for each media. Model architecture was written in TensorFlow<sup>71</sup> 1.14 using Python 3.6.7 with multiple open source libraries (citations, where relevant, are included in code for them). A mini-batch size of 1,024 was used for training and a mean squared error loss was optimized using a RMSProp optimizer<sup>76</sup> with a learning rate of 0.001. The stopping criterion monitored was the ‘r-squared’ value and the model was allowed to train for 10 epochs without improvement before stopping training. Training was carried out on a Google Cloud Tensor Processing Unit (TPU)<sup>72</sup> v3-8. Evaluation was carried out on 4 Tesla M60 GPUs. The model architecture visualization was generated using Netron 4.5.1. All processed data and models are publicly available on Zenodo at <https://zenodo.org/record/4436477> and all code is available on GitHub at [https://github.com/ledv/evolution/tree/master/manuscript\\_code/model/tpu\\_model](https://github.com/ledv/evolution/tree/master/manuscript_code/model/tpu_model). Transformer models are used in all the analyses in **Fig. 3** and **4** and **Extended Data Fig. 3-4, and 6-8**.

The models were tested by predicting expression on test sequences that the model had never seen before (**Supplementary Fig. 21**) that were measured in separate experiments, which included random, native (*i.e.* present in the yeast genome), and designed sequences. To obtain expression measurements for each tested sequence that are more accurate than those from the high-complexity training data experiment, library complexity was limited such that each test promoter sequence is observed in ~100 yeast cells (**Methods, Supplementary Information**).

### **Gene expression engineering using a genetic algorithm for sequence design**

To design<sup>77–80</sup> new sequences with desired expression, a genetic algorithm (GA) was implemented with the distributed evolutionary algorithms in python (DEAP) package<sup>81</sup>. The mutation probability and the two-point crossover probability were set to 0.1 and the selection tournament size was 3. The initial population size was 100,000 and the GA was run for 10 generations. The convolutional model was used as the basis for the objective function for GA, which was maximized for high expression and minimized for low expression (maximizing negative predicted expression). The top 500 sequences were synthesized (by Twist Biosciences) and expression was measured experimentally using our reporter assay, as described above.

### **Characterizing random genetic drift**

Simulation of random genetic drift (**Fig. 2a**) was initialized with a set of 5,720 random sequences, in generation 0. For each sequence in this starting set, a new single sequence was randomly picked from its  $3L$  mutational neighborhood (the set of all sequences at a Hamming distance of 1 from a sequence of length  $L$ ) and the difference in expression between the new sequence and the starting



sequence was calculated using the convolutional model (**Fig. 2b**). This was done for each starting sequence to get generation 1. Each subsequent generation  $n$  was produced by picking a single sequence randomly from the  $3L$  mutational neighborhood of each sequence in the preceding generation  $n-1$ . The simulation was carried out for 40 generations. Simulations were also subsequently repeated with the transformer model (**Extended Data Fig. 3f**), yielding concordant results.

For experimental validation, 1,000 random starting sequences were synthesized, introducing between one to three random mutations to these sequences. The expression levels of starting and mutated sequences were measured in both complex and defined media experimentally using our reporter assay. For 990 of these 1,000 starting sequences, experimental measurements were available for all three mutational distances. Additionally, 20 (median) separate single mutations were introduced to each of 196 native sequences, the sequences were synthesized, and their associated expression was measured similarly for both of these media; these were also included in the boxes for one mutational step in **Fig. 2c** and **Extended Data Fig. 1e**.

### **Characterizing the regulatory complexity of a sequence**

To estimate the regulatory complexity<sup>82,83</sup> of a sequence, the Gini coefficient of the regulatory interaction strengths for each TF was calculated. A new biochemical model was first trained with our defined media data to complement the existing one trained on complex media, using our published model architecture of TF binding and position-aware activity<sup>26</sup> and the training procedure previously described<sup>26</sup> (**Supplementary Notes**). The regulatory interaction strength was

then individually calculated for each regulator by setting the concentration parameter for that TF (individually) to 0 in the learned model, and the biochemical model was used to quantify the resulting change in expression, as previously described<sup>26</sup>. The resulting vector of interaction strengths was used to calculate a Gini coefficient for each sequence, separately for the complex and defined media models. The Gini coefficient is a measure of inequality of continuous values within a population, most commonly applied to wealth or income, and ranges from 0 (all members of the population have equal wealth) to 1 (the wealth of a population is held by a single individual). Regulatory complexity for a sequence is then 1-Gini, such that 1 indicates that all TFs contribute equally to the regulation of the gene and 0 indicates that a single TF is solely responsible for its regulation. As starting points for our trajectories, 200 native promoter sequences (from -160 to -80, relative to the TSS) were chosen with relatively high regulatory complexity and another 200 were chosen with relatively low regulatory complexity, spanning the range of predicted expression levels, as starting points for our trajectories.

Trajectories for stabilizing selection on gene expression were designed using the convolutional model (**Fig. 2d**). Here, all sequences were required to maintain a predicted expression level within 0.5 of the original expression levels at all steps along the trajectory. There was no explicit constraint on regulatory complexity in this simulation of stabilizing selection. In order to ensure that expression was unchanged, expression levels were measured experimentally for sequences along a trajectory at growing mutational steps from the initial sequence (2, 4, 8, 16, 32 mutations). Any trajectories for which an expression measurement was missing for *any* experimentally tested sequence were excluded from all analyses, retaining 172 trajectories with initial low regulatory complexity and 192 trajectories with initial high regulatory complexity. Testing whether observed

trends in regulatory complexity were affected by the degree to which expression was either predicted (by the convolutional model for 1-32 mutations) or observed (by the experiment at 2, 4, 8, 16, or 32 mutations) to be conserved, showed that the trends were robust to the degree of expression conservation (**Supplementary Fig. 11**).

### **Characterizing directional trajectories under SSWM**

Simulations of trajectories under a Strong Selection-Weak Mutation (SSWM)<sup>84-86</sup> regime were initialized with a set of native yeast promoter sequences (defined here as the subset from -160 to -80 relative to the TSS for all the genes in the yeast reference genome for which we had a good TSS estimate (Supplementary Table 3 in <sup>26</sup>) as the starting generation 0. For each sequence in generation  $n$ , the sequence from its  $3L$  mutational neighborhood that had the maximal (or separately, minimal) predicted expression using the convolutional model was picked to get generation  $n+1$ . The simulation was carried out for 10 rounds separately in the complex (**Fig. 2f**) and defined (**Extended Data Fig. 1f**) media. The simulations were subsequently repeated using the transformer model (**Extended Data Fig. 3i-j**).

For experimental validation, a subset of sequences from several generations were synthesized along mutational trajectories simulated by the convolutional model for complex media (10,322 sequences from 877 trajectories, 805 of which had every sequence along the trajectory successfully measured) and one for defined media (6,304 sequences from 637 trajectories, 591 of which had every sequence along the trajectory successfully measured) and their expression was measured in

the corresponding media experimentally using our reporter assay (**Fig. 2g, Extended Data Fig. 1g**).

### **Measuring the *URA3* expression-to-fitness relationship**

Two complementary environments were studied with opposite selective pressures on the expression of *URA3* (encoding an enzyme responsible for uracil synthesis): defined media, where organismal fitness increases with gene expression (up to saturation) and complex media + 5-FOA, where fitness decreases with Ura3p expression.

Convolutional models trained on defined and complex media were used to choose a set of 11 sequences that span a broad range of predicted expression levels in the two media when cloned into a YFP expression vector<sup>26</sup>. The relationship between expression of *URA3* and organismal fitness in yeast was estimated from experimental measurements with these 11 sequences, by cloning promoter sequence in front of YFP to measure expression level and in front of *URA3* to measure fitness. Unless otherwise noted, yeast were grown at 30°C, in an orbital shaker incubator at 225 RPM. Each vector was transformed into yeast (S288C::*ura3*), and three independent transformants were selected per vector to serve as biological replicates. For measuring expression, yeast were grown overnight in either YPD+NAT (yeast extract, peptone, dextrose, with 75µg/ml nourseothricin) or SD-Ura (synthetic defined media, lacking uracil; Sunrise Science 1703-500), and then re-inoculated in the morning and allowed to grow for 6 hours prior to measuring expression by flow cytometry for each replicate as the log ratio of YFP to the constant background RFP, including only cells obtaining the top 50% of RFP expression. Fitness was obtained by measuring the growth rate of each yeast strain in either SD-Ura or YPD+NAT+5-FOA (0.25 mg/ml 5-FOA). Yeast were grown continuously in triplicate in log phase, with linear shaking at 30°C in

a Synergy H1 plate reader (Biotek), by diluting each well to maintain  $OD < 0.7$ , with OD measured at 15 minute intervals. Growth rate was defined for each replicate as the median of the instantaneous smoothed growth rates over 5 measurements in log phase, considering only time points where  $0.05 < OD < 0.5$ . Each promoter's expression and growth rate were summarized as the mean of the three replicates.

### **Characterizing trajectories under conflicting expression objectives in different environments**

Simulations of sequence evolution in two complementary environments with opposite selective pressures (defined media and complex media) were initialized with a set of native yeast promoter sequences (present at -160 to -80 relative to the TSS) as the starting generation 0, with the objective function defined as the difference in predicted expression between defined and complex media (**Fig. 2h, Extended Data Fig. 2d-g**) using convolutional models trained in the respective media. The difference in expression between the two conditions was maximized at each iteration, which assumes that the cells are exposed to both environments before the mutations can reach fixation, an example of evolution in rapidly fluctuating environments<sup>31</sup>. For simplicity, it is assumed that fitness is directly proportional to higher expression in one condition and to lower expression in the other, such that mutations will be considered favorable even if they decrease fitness in one condition so long as they increase it in the other condition by a greater amount.

One simulation aimed to maximize the expression difference (defined minus complex), and the other to minimize it (maximizing complex minus defined). For each sequence in generation  $n$ , the sequence from its  $3L$  mutational neighborhood that had the maximum (or separately, minimum) value for the objective function based on the convolutional model prediction is picked for

generation  $n+1$ , to a total of 10 generations. The simulations were subsequently repeated using the transformer model yielding similar results (**Supplementary Fig. 17b-f**).

Motifs that were enriched in the sequences of generation 10 compared to the starting sequences were identified *de novo* using DREME<sup>87</sup>, and each of the top 5 consensus motifs were used as queries to search the YeTFaSCo database<sup>88</sup>, reporting the closest match, or one of multiple similar matches.

### **Finding orthologous promoters in the 1,011 *S. cerevisiae* genomes dataset**

To identify orthologs of S288C promoters in the whole genome sequences of the 1,011 yeast strains<sup>37</sup>, BLAT<sup>89</sup> was used to identify regions of  $\geq 80\%$  identity with each -160 to -80 region (relative to the TSS) annotated in the reference S288C genome sequence (R64)<sup>90</sup>. Any strains with more than one such match, where the match contained insertions or deletions, or had incomplete matches, were excluded on a gene-by-gene basis. Genes with more than 1.2 matches with  $\geq 80\%$  identity per genome, on average, were excluded altogether.

### **Computing the expression conservation coefficient (ECC)**

To calculate the ECC (a regulatory analog<sup>2,35,91,92</sup> of  $d_N/d_S$ <sup>34,93,94</sup>), for each yeast gene promoter, the transformer model was used to predict an expression value for each orthologous promoter in the 1,011 yeast genomes (above), defining an expression distribution with a standard deviation  $\sigma_B$ . Next, a set of sequences with random mutations was generated from each gene's consensus promoter sequence (defined as the most abundant base at each position across the strains), such

that the number of sequences at each Hamming distance from the consensus promoter sequence was the same for the natural and simulated sets. Here, mutations introduced to create random variation sampled each base with equal probability; using observed mutation rates yielded similar results (**Supplementary Information**). The same transformer model to predict the expression of the simulated sequences, and calculate its standard deviation  $\sigma_C$ . The nominal ECC is  $\log(\sigma_C/\sigma_B)$ . Because the variance on simulated sequences is better estimated than in natural orthologs (whose sequences may be more constrained), a constant correction factor is subtracted, calculated by creating a second simulated set of randomly mutated sequences whose diversity is limited to the same extent as in the natural set, by creating only one random mutation for every unique sequence in the set of native orthologs. Finally, the expression for this second set of sequences is predicted by the transformer model, and its standard deviation ( $\sigma_{C'}$ ) is used to calculate a null ECC for each gene ( $\log(\sigma_C/\sigma_{C'})$ ); the median of these null ECCs over all the genes is used as the constant correction factor  $C = \text{median}_{\forall \text{genes}, i} \left( \log_2 \left( \frac{\sigma_{C_i}}{\sigma_{C'_i}} \right) \right)$ . (An extensive description of the correction factor is provided in the “ECC calculation details and considerations” section of the **Supplementary Information**.)

The corrected ECC for gene  $g$  is then:

$$ECC_g = \log_2 \left( \frac{\sigma_{C_g}}{\sigma_{B_g}} \right) - C$$

The computed ECC values for all yeast genes, available in **Supplementary Table 1**, were used to identify cases of presumed stabilizing selection (selection favoring a fixed non-extreme value of a trait), diversifying (disruptive) selection (selection favoring more than one extreme values of a trait; as opposed to a single fixed intermediate value), and directional (positive) selection (selection favoring a single extreme value of a trait over all other possible values of the trait). Re-computing

the ECC values for all yeast genes using the S288C reference sequences instead of the consensus sequence for the promoters of each gene yielded very similar results.

In addition to each ECC value, a Z-score and *p*-values for the confidence that the observed ECC values differ from neutrality were also calculated. For each gene's true ECC, a set of matched random ECC values were calculated, where the denominator is a set of sequences matched for Hamming distance distribution and the total number of unique sequences. The null ECC mean and standard deviation were calculated from 1,111 such simulations, and used to calculate a Z-score for how extreme the actual ECC would be under this null distribution. This Z-score acts as a signed *p*-value (negative representing divergent expression and positive representing conservation), from which *p*-values (using the 'scipy.stats.norm.sf' function on the absolute value of the Z-score in Scipy<sup>95</sup> and multiplying the function's output by 2 to get a two-sided *p*-value) (**Supplementary Table 1**).

### **Inferring expression conservation across *Saccharomyces* species using RNA-seq data and comparing with ECC values**

Published RPKM values for orthologs of *S. cerevisiae* genes in closely related *Saccharomyces* species<sup>41</sup> were obtained from the Gene Expression Omnibus (GEO) (accession GSE83120). Only genes for which expression was quantified in all species were used in subsequent analysis. RPKM values were  $\log_2$  scaled after adding a pseudo count of 2, and the variance in expression of each gene across the species was calculated. Genes were ranked by their gene expression variance, and the 2% of genes with the lowest variance were considered as having conserved gene expression levels ('expression conserved'), while the 2% with the highest variance were considered



‘expression not-conserved’. The significance of the differences was robust to the choice of these thresholds (**Supplementary Information**). To compare to ECC values, the p-value of a two-sided Wilcoxon rank-sum test was estimated comparing the ECC values for genes in the ‘expression conserved’ and ‘expression not-conserved’ categories (implemented using the *scipy.stats.ranksums* SciPy<sup>95</sup> function). To control for the dependence between expression mean and variance, the analysis was repeated using the coefficient of variation ( $P = 1.05 \cdot 10^{-4}$ ) and the coefficient of dispersion ( $P = 2.42 \cdot 10^{-4}$ ) instead of variance, yielding similar results.

### **Experimental protocol for RNA-seq measurements from 11 Ascomycota species**

RNA-seq was performed on samples from the following 11 Ascomycota yeast species: *Saccharomyces cerevisiae*, *Saccharomyces bayanus*, *Naumovozyma (Saccharomyces) castellii*, *Candida glabrata*, *Kluyveromyces lactis*, *Kluyveromyces waltii*, *Candida albicans*, *Yarrowia lipolytica*, *Schizosaccharomyces japonicus*, *Schizosaccharomyces octosporus*, and *Schizosaccharomyces pombe*. Each of the 11 species was grown in BMW medium, chosen to minimize cross-species growth differences, as previously described<sup>96</sup>. *N. castellii* was grown at 25°C while the other species were grown at 30°C. RNeasy Midi or Mini Kits (Qiagen, Valencia, CA) were used to isolate total RNA from log-phase cells by mechanical lysis using the manufacturer instructions as previously described<sup>96</sup>. dUTP strand-specific RNA-seq libraries were constructed as previously described<sup>97</sup> with the following modifications. (1) The polyA<sup>+</sup>-selected RNA was fragmented in a 40µl reaction containing 1x Fragmentation Buffer (Affymetrix) by heating at 80°C for 4 minutes followed by cleanup via ethanol precipitation for all libraries (except *Y. lipolytica*, *S. pombe*, *S. japonicus*, and *S. octosporus*; for these species, the conditions described

previously were used<sup>97</sup>), followed by cleanup via 1.8x RNAClean XP beads (Beckman Coulter Genomics). **(2)** For *C. glabrata*, *K. lactis*, *S. bayanus*, *S. pombe*, *S. japonicus*, and *S. octosporus* libraries, the adapter ligation was performed overnight at 16°C. For the rest, this was done at 16°C for 2 hours as described previously<sup>97</sup>. **(3)** Normalization was carried out based on the cDNA input and pooling of selected Illumina barcoded-adaptor-ligated cDNA products followed by gel size selection occurred as follows: range of 275 to 575 bp for pooled *C. albicans*, *K. waltii*, and *N. castellii* libraries, and 375 to 575 bp for *C. glabrata*, *K. lactis*, and *S. bayanus* libraries. For the other libraries, no pooling was performed before gel size-selection – range of 310 to 510 bp for *Y. lipolytica* and 350 to 550 bp for *S. pombe*, *S. japonicus*, and *S. octosporus*. **(4)** The final PCR product was purified by 1.8x AMPure XP beads (Beckman Coulter Genomics) followed by a second gel size-selection for the range of 300 to 575 bp for *C. albicans*, *K. waltii*, and *S. castellii* libraries, but no second gel size-selection was performed for the other libraries. The pooled final library was sequenced on one to four lanes of HiSeq2000 (Illumina) with 68 base (*Y. lipolytica* had 76 base) paired-end reads and 8 base index reads.

## **Transcript assembly, mapping and expression calculation for the 11 Ascomycota species**

### **RNA-seq**

For each of the 11 Ascomycota yeast species above, reads were assembled using Trinity<sup>98</sup>(version ‘trinityrnaseq\_r2012-05-18’) and the assembled transcripts were mapped onto the assemblies to the respective genomes using GMAP<sup>99</sup>. The Jaccard coefficient was used to join adjacent assemblies given enough connecting reads (using the Trinity default of 0.35 for the Jaccard cutoff). Finally, upon mapping all assembled transcripts, the Jaccard coefficient was used to clip assemblies which did not have enough support over a certain region. For each of the species,

assembled transcripts were mapped to the genome sequence<sup>100</sup> using BLAT<sup>89</sup>. Estimated expression values were calculated for each transcript using RSEM<sup>101</sup> (defined in RSEM as the estimate of the number of fragments that are derived from a given isoform or gene, or the expectation of the number of alignable and unfiltered fragments that are derived from an isoform or gene given the maximum likelihood abundances). Only reads mapping to the sense mRNA strand were considered. Orthology between genes in different species was used as previously described<sup>100</sup>.

### **Inferring expression conservation across Ascomycota species using our RNA-seq data and comparing with ECC values**

Estimated expression values from the 11 Ascomycota species RNA-seq data were used after removing all genes with NA values in expression for more than three species. Estimated expression values were  $\log_2$  scaled after adding a pseudo count of 1, and the variance in expression for each gene across the species was calculated. Genes were ordered by their variance in expression across the reported fungal species. Here, the 10% of genes with the lowest expression variance were considered to have ‘conserved’ expression, and the 10% with highest expression variance were considered to have expression ‘not conserved’. To compare to ECC values, the p-value of a two-sided Wilcoxon rank-sum test was estimated comparing the ECC values for genes in the ‘conserved’ and ‘not conserved’ categories (implemented using the *scipy.stats.ranksums* SciPy<sup>95</sup> function). Similar results were obtained when repeating the analysis using the coefficient of variation ( $P = 4.22 \cdot 10^{-5}$ ) and the coefficient of dispersion ( $P = 8.05 \cdot 10^{-5}$ ) instead of variance.

## **Inferring expression conservation across mammalian species using RNA-seq data and comparing with ECC values**

Ensembl Biomart<sup>102</sup> was used to find one to one orthologs of *S. cerevisiae* genes in humans (of 'Human homology type' 'ortholog\_one2one'; all 'ortholog\_one2many' and 'many2many' orthologs were excluded). For these human orthologs of yeast genes, the previously reported 'evolutionary variance' values across mammalian species from the original publication<sup>42</sup> (based on an Ornstein Uhlenbeck (OU model)<sup>42</sup>) were directly used. Here, the 25% of genes with the lowest 'evolutionary variance' were considered to have conserved expression and the top 25% were considered to be not conserved (the same thresholds used in the original study<sup>42</sup>). This was done separately for each profiled tissue (brain, heart, kidney, liver, lung and skeletal muscle). Subsequently, a human ortholog for a yeast gene was considered to have conserved (or non-conserved) expression if it was found to have conserved (or non-conserved) expression in at least one of the profiled tissues. Genes with conflicting expression conservation classes across tissues were excluded from the analysis. To compare to ECC values, the p-value of a two-sided Wilcoxon rank-sum test was estimated comparing the ECC values for genes in the "conserved" and "not conserved" categories (implemented using the *scipy.stats.ranksums* SciPy<sup>95</sup> function).

## **Quantifying sequence dissimilarity using mean Hamming distance**

For each group of orthologous yeast gene promoters (with ungapped alignments), the mean of Hamming distances between each pair of orthologous promoters across the 1,011 isolates was calculated.

## **Generation of CDC36 promoter strains by allele swapping**

Strains with a restored Upc2p binding site in the *CDC36* promoter region were obtained using a previously described CRISPR-Cas9 method<sup>103</sup>. Guide RNAs (gRNAs) were designed using the Benchling online tool (<https://www.benchling.com/>) and cloned in a pGZ110 derived plasmid<sup>104</sup>, using standard “Golden Gate Assembly”<sup>105</sup>. Plasmids carrying the gRNA and Cas9 gene were then co-transformed with a synthetic DNA fragment (ssODN) composed of a 100 bp sequence with perfect complementarity to the background promoter sequence (WE) but for the centrally-located targeted alleles that overlap the Upc2p binding site. Allele swapping was confirmed by Sanger sequencing (Macrogen, South Korea). Sequences were analyzed using the SGRP (Saccharomyces Genome Resequencing Project) BLAST server ([http://www.moseslab.csb.utoronto.ca/sgrp/blast\\_new/](http://www.moseslab.csb.utoronto.ca/sgrp/blast_new/)) and MUSCLE tool in Geneious v10.1. All primers and ssODNs used are listed in **Supplementary Table 2**.

### **RNA extraction and qPCR of *CDC36***

Gene expression analysis was performed by qPCR from cultures growth in SD medium supplemented with uracil (0.02% p/v). Samples were grown until exponential phase (OD 0.6-0.8), collected by centrifugation and treated with 10 units of Zymolyase 20T (50mg/ml) for 30 min at 37°C. RNA was extracted using E.Z.N.A Total RNA kit I (OMEGA) according to manufacturers’ instructions. Genomic DNA traces were then removed by treating samples with DNase I (Promega). RNA concentrations were estimated using a Qubit system and verified by 1.5% agarose gel. RNA extractions were performed in three biological replicates.

cDNA was synthesized using 200 units of M-MLV Reverse transcriptase (Promega), 0.5  $\mu\text{g}$  of Oligo (dT)15 primer and 1  $\mu\text{g}$  of RNA in a final volume of 25  $\mu\text{L}$  according to manufacturers' instructions. qPCR reactions were carried out using Brilliant II SYBR® Green QPCR Master Mix (Agilent Technologies) in a final volume of 10  $\mu\text{L}$ , containing 0.2  $\mu\text{M}$  of each primer and 1  $\mu\text{L}$  of the cDNA previously synthesized. qPCR reactions were carried out in three technical replicates per biological replicate using an Eco Real-Time PCR system (Illumina, Inc.) under the following conditions: 95°C for 15 min and 40 cycles at 95°C for 10 s and 58°C for 30 s. Primers used are listed in **Supplementary Table 2**. The relative expression of *CDC36* was quantified using the  $2^{-\Delta\Delta\text{Ct}}$  approach<sup>106</sup>, and normalized with two housekeeping genes as previously described<sup>107</sup>, using the median Ct of the three technical replicates for each sample. The housekeeping genes *ACT1* and *RPN2* were used as previously described<sup>108</sup>.

### **Growth curves of *CDC36* mutant and wild type alleles**

Growth curves incorporating carbon source switching from glucose to galactose were generated as previously described<sup>109</sup>. Pre-cultures were grown in YNB containing 5% glucose medium at 30°C for 24 h. Cultures were then diluted to an initial OD<sub>600nm</sub> of 0.1 in fresh YNB 5% glucose medium for an extra overnight growth. The next day, cultures were used to inoculate a 96-well plate with a final volume of 200 $\mu\text{L}$  YNB with 5% galactose with an initial OD<sub>600nm</sub> of 0.1. In parallel, a control plate containing YNB with 5% glucose was similarly inoculated. All experiments were performed in triplicate. OD<sub>600nm</sub> was monitored every 30 min using a Tecan Sunrise absorbance microplate reader (Tecan Group Ltd.). The kinetic parameters of lag phase, growth efficiency ( $\Delta\text{OD}_{600\text{ nm}}$ ) and maximum specific growth rate ( $\mu_{\text{max}}$ ) were determined as previously described<sup>110</sup>, fitting the curves with the Gompertz function using R version 3.3.2. All

growth parameters are expressed as the ratio of growth within YNB+galactose to YNB+glucose to control for phenotypic variation that results from something other than the carbon source switch.

### **Fitness responsivity**

The empirically-determined relationships between the expression levels to organismal fitness for each of 80 genes<sup>11</sup> were re-analyzed. Published expression-to-fitness curves in glucose media for each of 80 genes were obtained from the Supplementary Data of the original publication<sup>11</sup>. For each of these curves, the total variation (**Extended Data Fig. 5**) was calculated by partitioning the expression range into 36 regular intervals (as reported in the ‘impulse fit’ of the expression-to-fitness curves in the original publication<sup>11</sup>) and summing the absolute difference in fitness at the endpoints of each partition as follows  $\sum |F_{GENE}(e_{i+1}) - F_{GENE}(e_i)|$ , for each gene’s expression-to-fitness function,  $F_{GENE}(e)$ . The same qualitative relationship between a gene’s ECC and fitness responsivity as reported in other studies<sup>111–113</sup> was observed, including *LCB2* (ECC 2.15 and high fitness responsivity<sup>112</sup>) and *MLS1* (ECC -1.32 and extremely low fitness responsivity<sup>113</sup>).

### **Mutational robustness**

For every sequence, mutational robustness was defined as the fraction of sequences in its *3L* mutational neighborhood that altered the expression by an amount less than  $\epsilon$ , where  $\epsilon$  is set at two times the standard deviation of expression variance across all genes with an ECC >0 (here,  $\epsilon = 0.1616$ ; ECC calculated using the 1,011 *S. cerevisiae* genomes, **Extended Data Fig. 4c**). Using different values for  $\epsilon$  yielded very similar results.

## The evolvability vector

To derive the evolvability vector for a given sequence, expression changes associated with single base changes in every possible position were sorted to obtain a monotonically increasing vector of length  $3L$  for each sequence of length  $L$  (here,  $L=80$ ;  $3L=240$ ; **Fig. 4a**, left, **Methods**). Formally, to compute an evolvability vector for a sequence  $s_0$ , for each sequence  $s_i$  in the  $3L$  mutational neighborhood of  $s_0$ , the difference between the predicted expression of  $s_i$  and that of  $s_0$  :  $d_i = f(s_i) - f(s_0)$  was calculated, where  $f(s)$  represents the predicted expression of the transformer model. The evolvability vector is defined as the vector  $D(\{d_1, d_2, \dots, d_{3L}\})$ , sorted such that  $d_i \geq d_{i-1}, \forall i$  (i.e.  $d_i$  values are in ascending order).

## Power law distribution analysis

The list of the absolute values of the evolvability vectors for all native sequences was used to define the distribution of the magnitude of the expression effect of mutations. The powerlaw<sup>114</sup> Python package was used to determine whether the data fit a power law distribution. The ‘Fit’ function with an ‘xmin’ parameter of 0.5 was used to determine the exponent and the ‘distribution\_compare’ function was used to determine the p-value for the fit (**Extended Data Fig. 4d, Supplementary Fig. 2h**).

## Characterizing the archetypal evolvability space

The evolvability vectors for a new random sample of a million sequences were used as input to an autoencoder with an archetypal regularization constraint<sup>45</sup> on the embedding layer. The autoencoder was trained using the AANet implementation made available with the publication<sup>45</sup>



with no noise added to the archetypal layer during training, a linear activation on the output layer, an equal weight of 1 on each of the loss terms (the mean squared error loss term along with the non-negativity and convexity constraints), a learning rate of 0.001, and a minibatch size of 4,096. The autoencoder accepts an evolvability vector (of length 240 for an 80bp sequence) as input to the first encoder layer, where each node in the input layer is connected to each node in the encoder layer (fully connected layer). Every layer in the autoencoder was fully connected. The encoder architecture used was [1024, 512, 256, 128, 64], where each entry corresponds to the number of nodes in the corresponding hidden layer and the decoder architecture was the encoder's mirror image. The output layer was the same shape as input layer and each node in the last decoder layer was connected to each node in the output layer. To select the optimal number of archetypes, the autoencoder was first trained for a 1,000 minibatches separately for 1 to 9 archetypes. Following the recommended approach<sup>45</sup> for picking the optimal number of archetypes, we used an elbow plot of mean squared error on the evolvability vectors (here, using native sequences) vs. the number of archetypes in the autoencoder (**Extended Data Fig. 6a**).

The autoencoder was then trained from scratch with 3 archetypes, using the full training data and parameters for 250,000 batches. Since this autoencoder aims to reconstruct the original evolvability vector for each sequence by learning feature representations after passing them through an information bottleneck, its reconstruction accuracy was first verified on the set of native yeast promoter sequences (**Extended Data Fig. 6b**, Pearson's  $r = 0.992$ ). To visualize the evolvability vectors corresponding to sequences in 2 dimensions (2D), the evolvability vectors corresponding to the three archetypes were first generated by decoding their archetypal latent space coordinates ((1,0,0), (0,1,0) and (0,0,1)) through the decoder, and MDS was performed on the

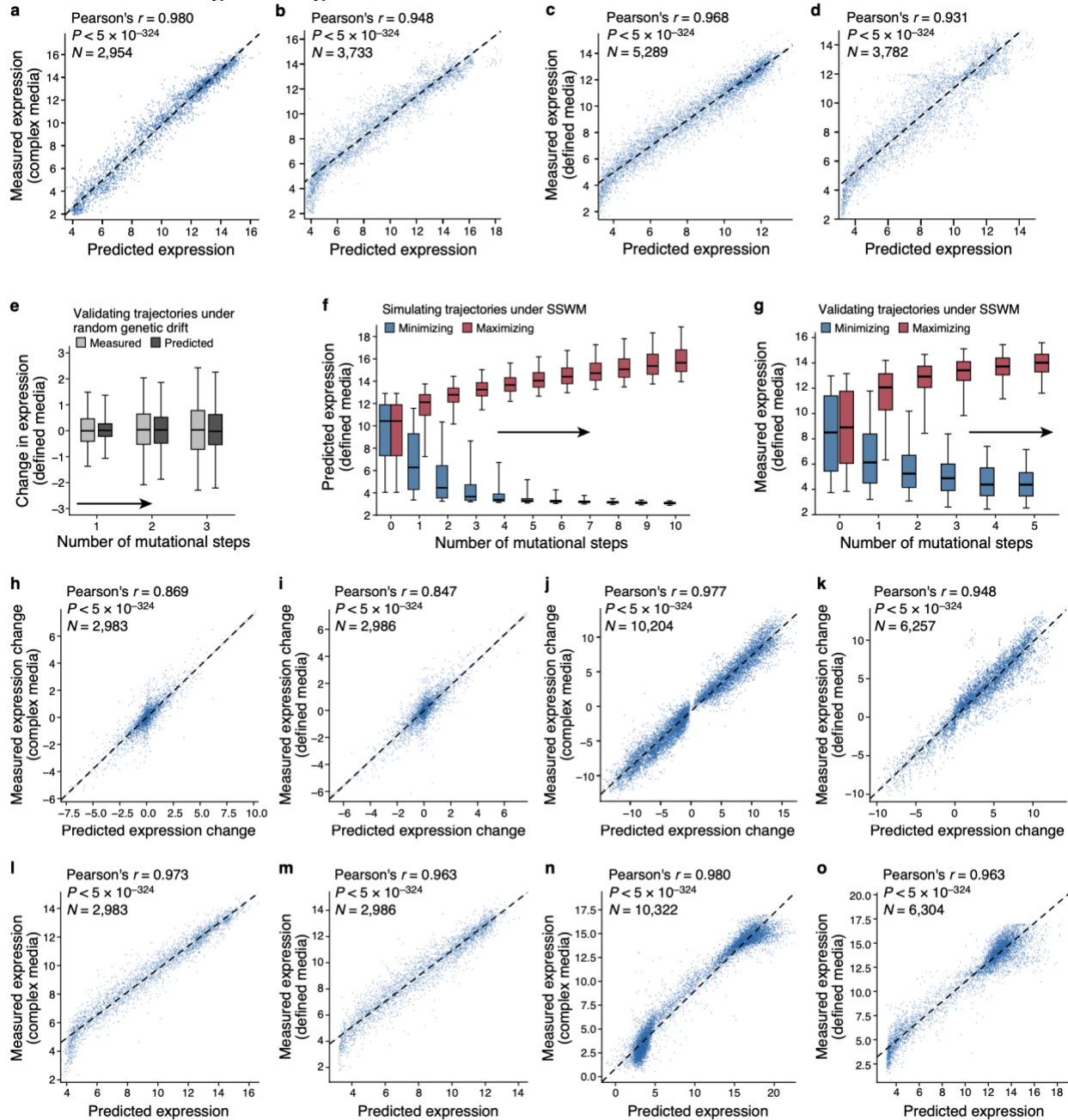
decoded evolvability vectors of the archetypes. Then, as previously described<sup>45</sup>, the encoded evolvability vector of each new sequence was projected into the 2D MDS space by representing it as a mixture of the archetypes and interpolating them between the MDS coordinates of each archetype. For every sequence, the following equivalent representations can now be computed: (i) its evolvability vector, (ii) an archetypal triplet quantifying the similarity of its encoded (latent space) evolvability vector to the three archetypes and (iii) a two-dimensional multidimensional scaling (MDS) coordinate<sup>45</sup> for visualizing the evolvability vectors. The representation of the evolvability vector for each sequence in this archetypal space is now bounded by a simplex (whose vertices correspond to the 3 evolvability archetypes). For each native and natural yeast promoter sequence from the sequence space, the archetypal triplet and MDS coordinates were inferred using its evolvability vector with this trained autoencoder. The MDS coordinates for the archetypes and the native yeast promoter sequences were used to generate the visualizations of the sequence space shown. This archetypal characterization of evolvability vectors allows the encoding and visualization of sequences by their evolvability in the context of a fitness landscape.

### **Visualizing promoter fitness landscapes**

1000 random sequences were sampled and projected onto the MDS coordinate system for visualizing the sequence space described above. The expression level of each sequence was calculated using our model, and expression values were scaled so that the minimum was 0 and maximum was 1. Previously quantified expression-to-fitness relationships<sup>11</sup> to compute fitness (fraction of wildtype growth rate) by using cubic spline interpolation (implemented using the *scipy.interpolate.CubicSpline* SciPy<sup>95</sup> function) on the expression level after scaling the measured expression-to-fitness curves to have an expression range of 0 to 1. These fitness values were then

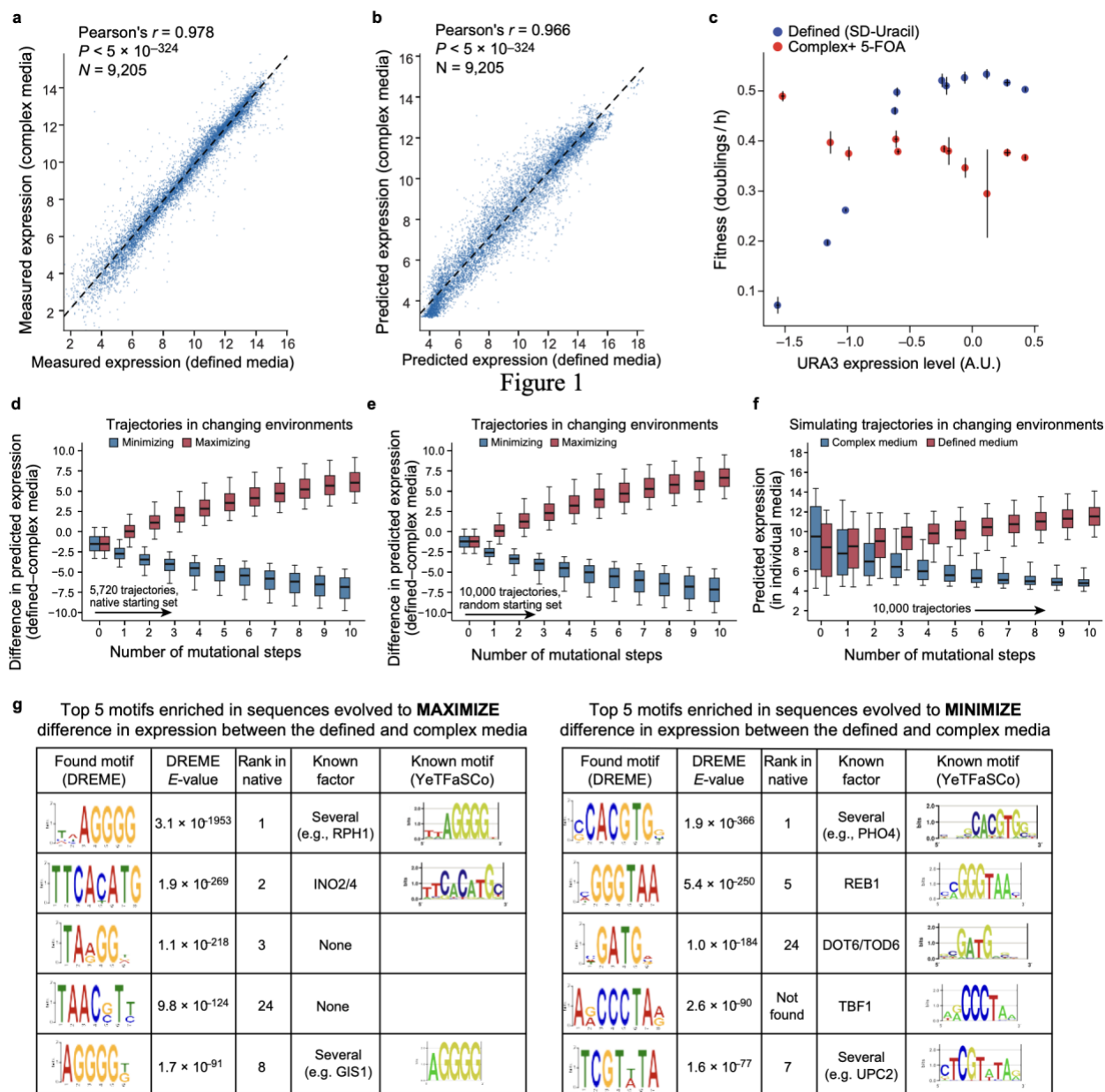
used to generate the contour plots (implemented using the *matplotlib.pyplot.tricontourf* function; **Fig. 4e, Extended Data Fig. 7**) that visualize the fitness landscape in that gene's promoter sequence space.

## Extended Data Figures Legends



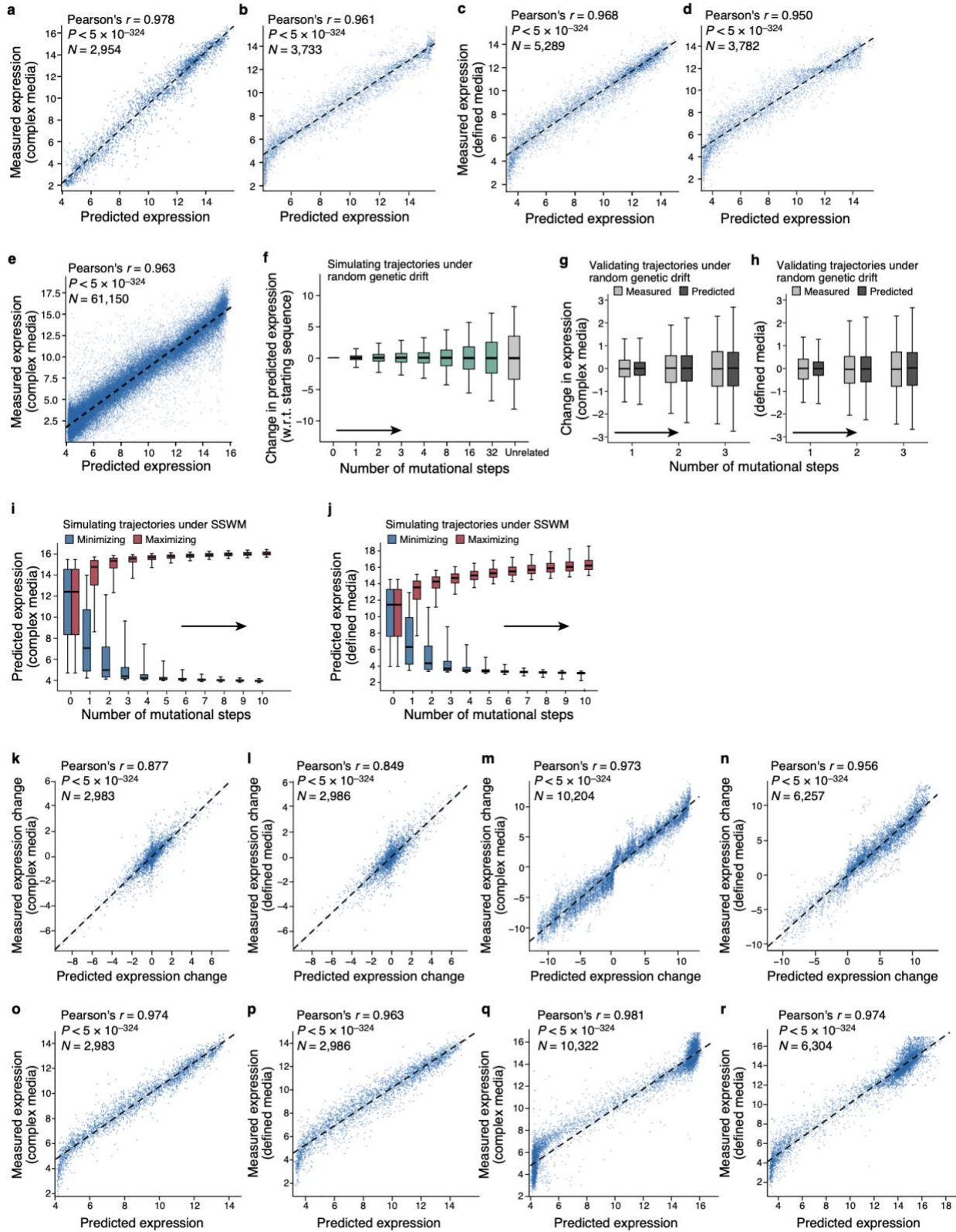
**Extended Data Fig. 1: The convolutional sequence-to-expression model generalizes reliably and helps characterize sequence trajectories under different evolutionary regimes. (a-d)** Prediction of expression from sequence in complex (YPD) (a-b) and defined (SD-Uracil) (c-d) media. Predicted (x axis) and experimentally measured (y axis) expression for (a,c) random test sequences (sampled separately from and not overlapping with the training data) and (b,d) native yeast promoter sequences containing random single base mutations. Top left: Pearson's  $r$  and associated two-tailed p-value. Compression of predictions in the lower left results from binning differences during cell sorting in different experiments (**Supplementary Notes**). **e**, Experimental validation of trajectories from simulations of random genetic drift. Distribution of measured (light grey) and predicted (dark gray) changes in expression in the defined media (SD-Uracil) (y

axis) for the synthesized randomly-designed sequences (n=2,986) at each mutational step (x axis). Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **f, g**, Simulation and validation of expression trajectories under SSWM in defined media (SD-Uracil). **f**, Distribution of predicted expression levels (y axis) in defined media at each evolutionary time step (x axis) for sequences under SSWM favoring high (red) or low (blue) expression, starting with native promoter sequences (n=5,720). Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **g**, Experimentally-measured expression distribution in defined media (y axis) for the synthesized sequences (n=6,304 sequences; 637 trajectories) at each mutational step (x axis) from predicted mutational trajectories under SSWM, favoring high (red) or low (blue) expression. Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **h-o**, Experimental validation of predicted expression for sequences from the random genetic drift and SSWM simulations. Experimentally measured (y axis) and predicted (x axis) expression level (**l-o**) or expression change from the starting sequence (**h-k**) in complex (**h,j,l,n**) or defined (**i,k,m,o**) media using sequences from the random genetic drift (**Fig. 2c** and **(e)**; **h,i,l,m** here) and SSWM (**Fig. 2g** and **(g)**; **j,k,n,o** here) validation experiments. Top left: Pearson's *r* and associated two-tailed p-values.



**Extended Data Fig. 2 | Characterization of sequence trajectories under strong competing selection pressures using the convolutional model. a,b**, Expression is highly correlated between defined and complex media. Measured (**a**) and predicted (**b**) expression in defined ( $x$  axis) and complex ( $y$  axis) media for a set of test sequences measured in both media. Top left: Pearson's  $r$  and associated two-tailed  $p$ -values. **c**, Opposing relationships between organismal fitness and *URA3* expression in two environments. Measured expression ( $x$  axis, using a YFP reporter) and fitness ( $y$  axis; when used as the promoter sequence for the *URA3* gene) for yeast with each of 11 promoters predicted to span a wide range of expression levels in complex media with 5-FOA (red), where higher expression of *URA3* is toxic due to *URA3*-mediated conversion of 5-FOA to 5-fluorouracil, and in defined media lacking uracil (blue), where *URA3* is required for uracil synthesis. Error bars: Standard error of the mean ( $n=3$  replicate experiments). **d-f**, Competing expression objectives are slow to reach saturation. **d,e**, Difference in predicted expression ( $y$  axis) at each evolutionary time step ( $x$  axis) under selection to maximize (red) or

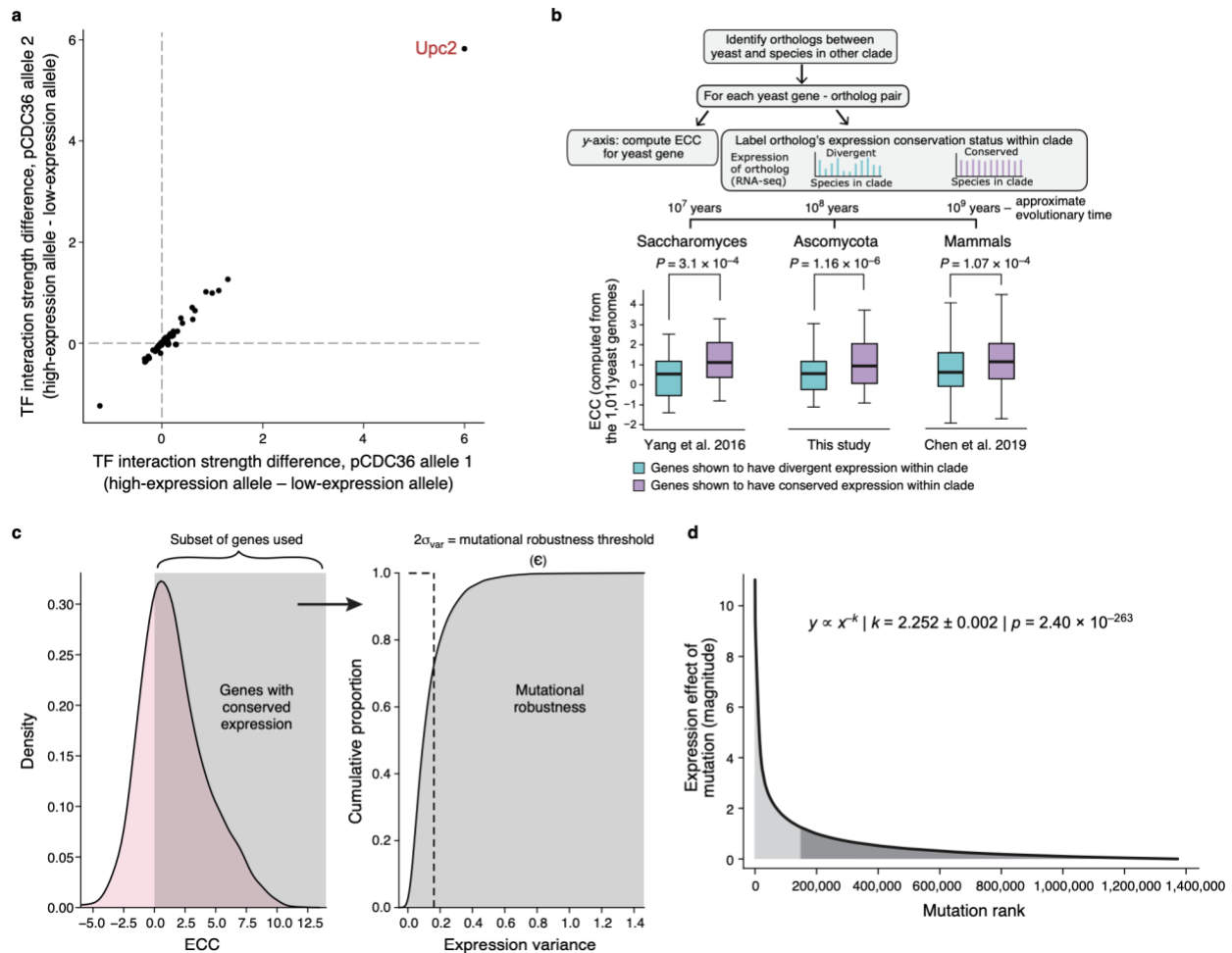
minimize (blue) the difference between expression in defined and complex media, starting with either native sequences (**d**, as **Fig. 2h**,  $n=5,720$ ) or random sequences (**e**,  $n=10,000$ ). **f**, Distribution of predicted expression ( $y$  axis) in complex (blue) and defined (red) media at each evolutionary time step ( $x$  axis) for a starting set of random sequences ( $n=10,000$ ). Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **g**, Motifs enriched within sequences evolved for competing objectives in different environments. Top five most enriched motifs, found using DREME<sup>87</sup> (**Methods**) within sequences computationally evolved from a starting set of random sequences to either maximize (left) or minimize (right) the difference in expression between defined and complex media, along with DREME E-values, the corresponding rank of the same motif when using native sequences as a starting point, the likely cognate TF and that TF's known motif.



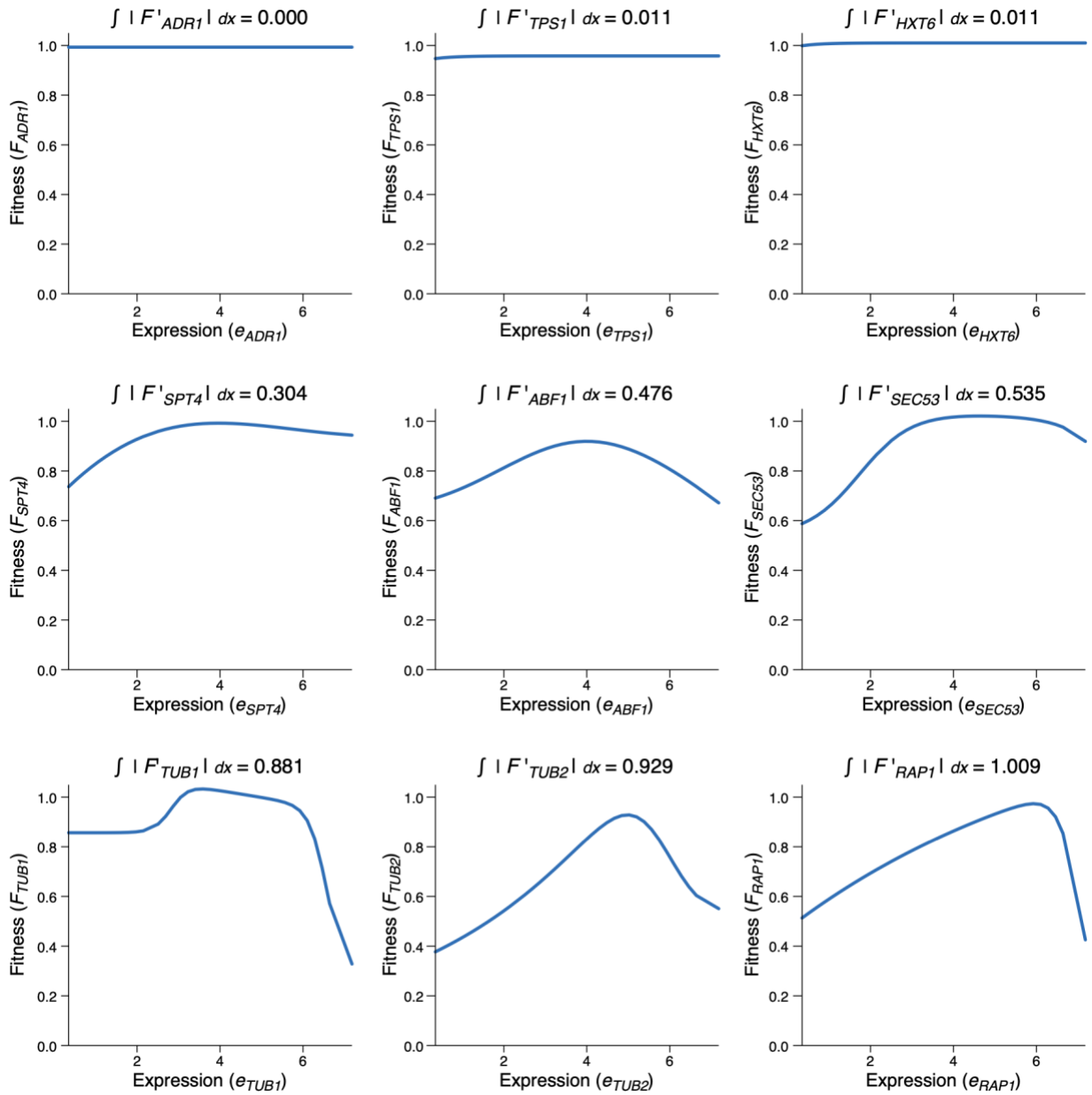
**Extended Data Fig. 3 | The transformer sequence-to-expression model generalizes reliably and helps characterize sequence trajectories under different evolutionary regimes. a-d,** Prediction of expression from sequence in the complex (a-b) and defined (c-d) media. Predicted



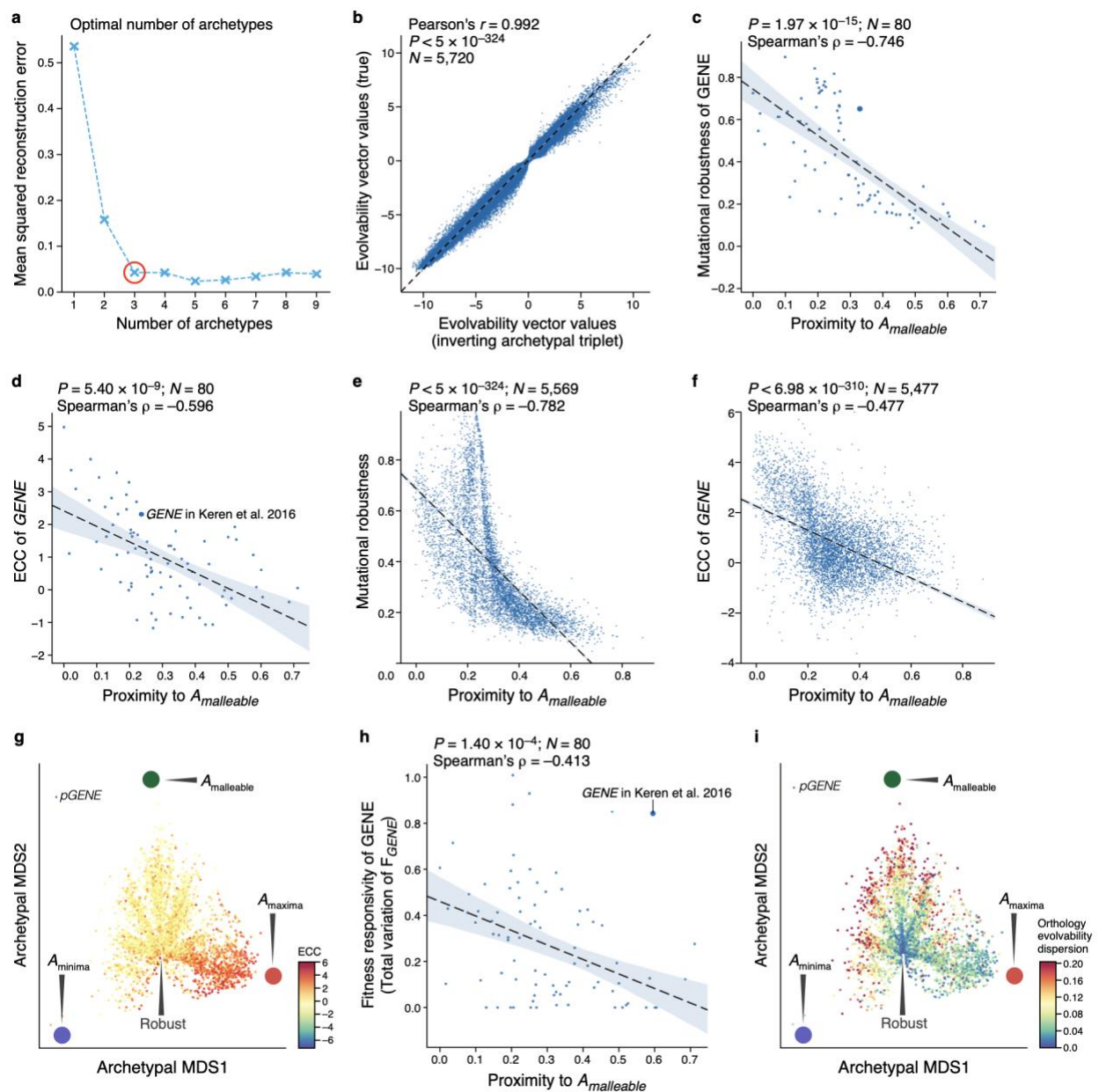
(*x* axis) and experimentally measured (*y* axis) expression for **(a,c)** random test sequences (sampled separately from and not overlapping with the training data) and **(b,d)** native yeast promoter sequences containing random single base mutations. Top left: Pearson's *r* and associated two-tailed *p*-value. Compression of predictions in the lower left results from binning differences during cell sorting in different experiments (**Supplementary Notes**). **e**, Predicted (*x* axis) and experimentally measured (*y* axis) expression in complex media (YPD) for all native yeast promoter sequences. Pearson's *r* and associated two-tailed *p*-values are shown. **f**, Predicted expression divergence under random genetic drift. Distribution of the change in predicted expression (*y* axis) for random starting sequences (*n*=5,720) at each mutational step (*x* axis) for trajectories simulated under random genetic drift. Silver bar: differences in expression between unrelated sequences. **g,h**, Comparison of the distribution of measured (light grey) and transformer model predicted (dark grey) changes in expression (*y* axis) in complex media (**g**, *n*=2,983) and defined media (**h**, *n*=2,986) for synthesized randomly-designed sequences at each mutational step (*x* axis). **i,j** Predicted expression evolution under SSWM. Distribution of predicted expression levels (*y* axis) in complex media (**i**, *n*=10,322) and defined media (**j**, *n*=6,304) at each mutational step (*x* axis) for sequence trajectories under SSWM favoring high (red) or low (blue) expression, starting with 5,720 native promoter sequences. **(f-j)** Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **k-r**, Comparison of model predicted expression for sequences synthesized previously for the random genetic drift and SSWM analyses. Experimentally measured (*y* axis) and transformer model predicted (*x* axis) expression level (**o-r**) or expression change from the starting sequence (**k-n**) in complex (**k,m,o,q**) or defined (**l,n,p,r**) media using sequences from the random genetic drift (**Fig. 2c** and **(Extended Data Fig. 1e)**; **k,l,o,p** here) and SSWM (**Fig. 2g** and **(Extended Data Fig. 1g)**; **m,n,q,r** here) validation experiments. Top left: Pearson's *r* and associated two-tailed *p*-values.



**Extended Data Fig. 4 | Signatures of stabilizing selection on gene expression detected from regulatory DNA across natural populations.** **a**, Expression-altering alleles in the CDC36 promoter are attributed primarily to altered UPC2 binding. TF interaction strength<sup>26</sup> (expression attributable to each TF) difference between the high and low alleles (each point is a TF) for each of two low expression alleles (allele 1:  $x$  axis; allele 2:  $y$  axis). Each low-expressing allele is compared to the high-expression allele with the most similar sequence (across all promoter sequences analyzed from the 1,011 strains;  $e_{TF,A_{high}} - e_{TF,A_{low}}$ ). **b**, Distribution of ECC ( $y$  axis, calculated from 1,011 *S. cerevisiae* genomes, top left) for *S. cerevisiae* genes whose orthologs have divergent (blue) or conserved (purple) expression (within *Saccharomyces* (left,  $n=4,191$ ), Ascomycota (middle,  $n=4,910$ ), or mammals (right,  $n=199$ ) (as determined by cross species RNA-seq, top right).  $p$ -values: two-sided Wilcoxon rank-sum test. Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **c**, Determination of expression change threshold for defining a "tolerated mutation" to compute mutational robustness. We used all genes with an ECC consistent with stabilizing selection ( $ECC > 0$ ; left), calculated the variance in predicted expression across the 1011 yeast strains for each gene, and chose the tolerable mutation threshold,  $\epsilon$ , as two standard deviations of the distribution of the variance (right). ~73% of genes with  $ECC > 0$  had an expression variation lower than  $\epsilon$ . **d**, Distribution of the effects of mutations (magnitude) on expression for all native regulatory sequences follows a power law with an exponent of 2.252. Shaded regions are equal in area.

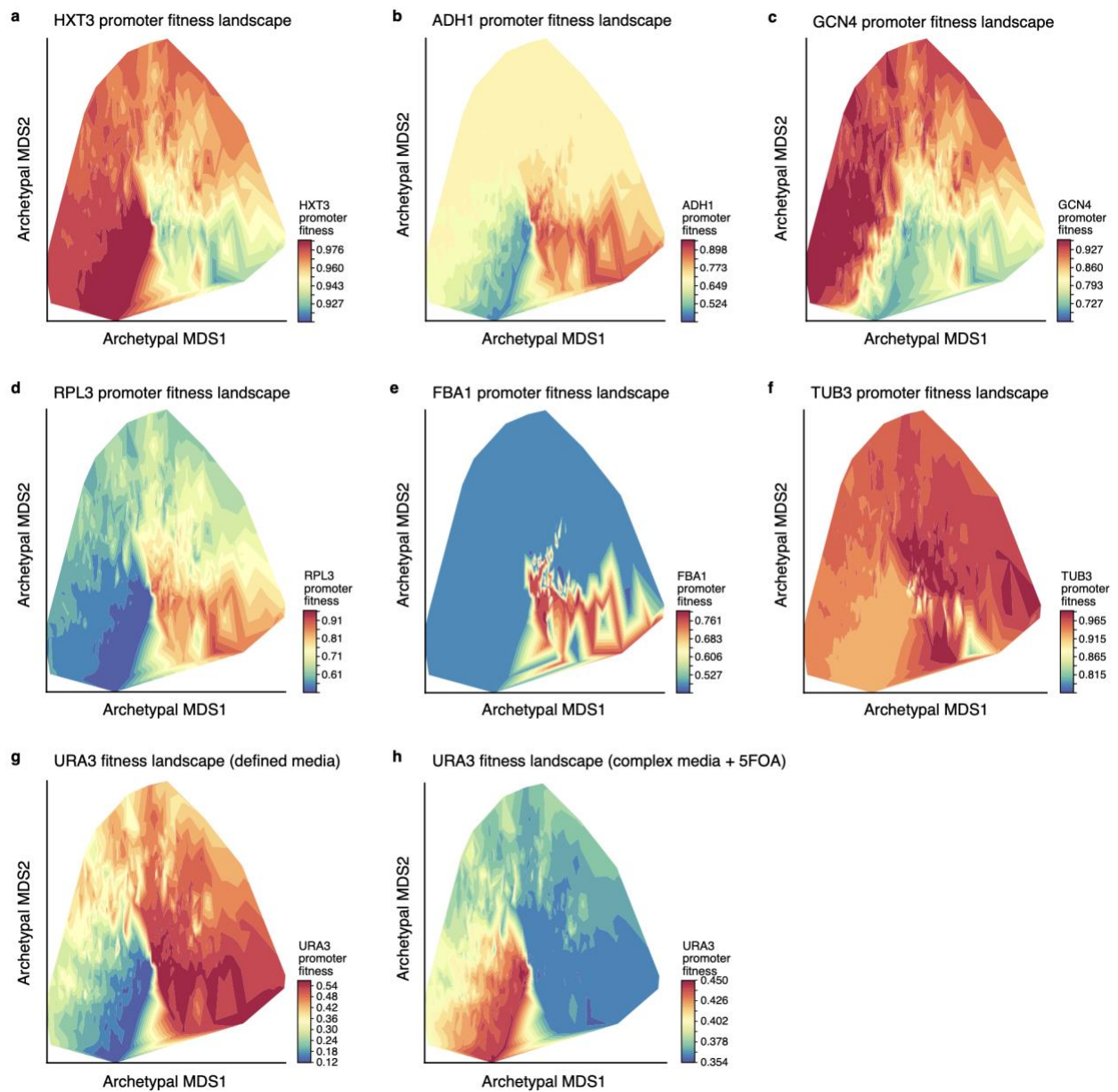


**Extended Data Fig. 5 | Fitness responsivity of a gene as the total variation of its expression-to-fitness relationship  $F_{GENE}$  curves.** Expression ( $x$  axis) and fitness ( $y$  axis) levels for different promoter variants for each select gene fit from experimental measurements by Keren et al<sup>11</sup>. Fitness responsivity calculated as the total variation in each curve is noted above each panel.



**Extended Data Fig. 6 | Analysis of regulatory evolvability reveals sequence-encoded signatures of expression conservation from solitary sequences.** **a**, Selection of optimal number of archetypes. Mean-square-reconstruction error ( $y$  axis) for reconstructing the evolvability vectors from the embeddings learned by the autoencoder for an increasing number of archetypes ( $x$  axis). Red circle: optimal number of archetypes selected as prescribed<sup>45</sup> by the “elbow method”. **b**, The archetypal embeddings learned by the autoencoder accurately capture evolvability vectors. Original ( $y$  axis) and reconstructed ( $x$  axis) expression changes (the values in the evolvability vectors) for each native sequence (none seen by the autoencoder in training). Top left: Pearson’s  $r$  and associated two-tailed  $p$ -values. **c-f**, Evolvability space captures regulatory sequences’ evolutionary properties. Proximity to the malleable archetype ( $A_{\text{malleable}}$ ) ( $x$  axis) and mutational robustness (**c,e**  $y$  axis) or ECC (**d,f**  $y$  axis) for all yeast genes (**e,f**) or the gene for which fitness responsiveness was quantified (**c,d**). Top right: Spearman’s  $\rho$  and associated two-sided  $p$ -value. “L”-shape of relationship in **e** results from the robust cleft,  $A_{\text{maxima}}$ , and

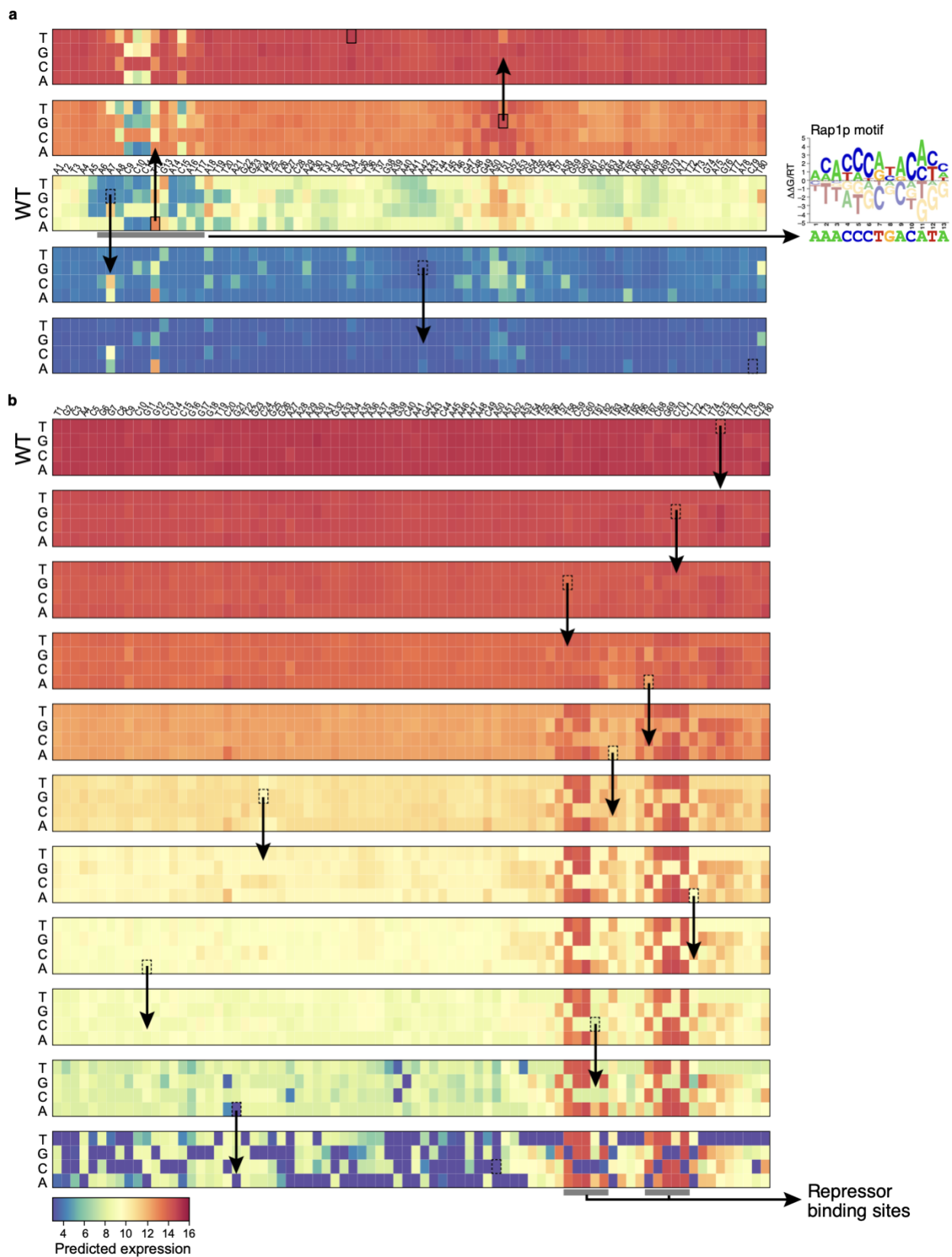
$A_{\text{minima}}$  all being distal to  $A_{\text{malleable}}$  (left side of plot). **g**, All native (S288C reference) promoter sequences (points) projected onto the archetypal evolvability space learned from random sequences; colored by their ECC. Large colored circles: evolvability archetypes. **h**, The proximity to the malleable archetype ( $x$  axis) and fitness responsiveness ( $y$  axis) for the 80 genes with measured fitness responsiveness. Top right: Spearman's  $\rho$  and associated two-tailed  $p$ -values. Light blue error band: 95% confidence interval. **i**, All native (S288C reference) promoter sequences (points) projected on the evolvability space learned from random sequences; colored by their mean pairwise distance in the archetypal evolvability space between all promoter alleles across the 1,011 yeast isolates for that gene (ortholog evolvability dispersion). Large colored circles: evolvability archetypes.



**Extended Data Fig. 7 | Visualizing promoter fitness landscapes in sequence space.**

Visualizing the fitness landscapes for the promoters of *HXT3* (a), *ADH1* (b), *GCN4* (c), *RPL3* (d), *FBA1* (e), *TUB3* (f), *URA3* (in defined media) (g), *URA3* (in complex media + 5FOA) (h).

1000 promoter sequences represented by their evolvability vectors projected onto the 2D archetypal evolvability space and colored by their associated fitness as reflected by their predicted growth rate relative to wildtype (color, **Methods**), estimated by first mapping sequences to expression with our model and then expression to fitness as measured and estimated previously<sup>11</sup>.



**Extended Data Fig. 8 | In silico mutagenesis (ISM) of malleable and robust promoters.** SSWM trajectories for (a) *DBP7*, a malleable promoter, and (b) *UTH1*, a robust promoter. Each

subplot shows the *in silico* mutagenesis effects for how expression level (color) changes when mutating each position (*x* axis) to each of the four bases (*y* axis) of each sequence (subplots) in the trajectories. The DNA sequence is indicated above each wildtype subplot (indicated with “WT” at left). Arrows indicate the mutations selected at each step, which always correspond to the mutation of maximal effect; increasing expression goes up the figure from wildtype and decreasing expression goes down. Part of the malleability of the *DBP7* promoter results from an intermediate-affinity Rap1p binding site (gray bar). The first mutations in increasing- and decreasing-expression trajectories either increase or decrease (respectively) the affinity of this site. The *UTH1* promoter changes gradually in expression and evolves proximal repressor binding sites to dampen expression (gray bars).



## References

1. Wittkopp, P. J. & Kalay, G. Cis-regulatory elements: molecular mechanisms and evolutionary processes underlying divergence. *Nat. Rev. Genet.* **13**, 59–69 (2011).
2. Hill, M. S., Vande Zande, P. & Wittkopp, P. J. Molecular and evolutionary processes generating variation in gene expression. *Nature Reviews Genetics* 1–13 (2020).
3. Fuqua, T. *et al.* Dense and pleiotropic regulatory information in a developmental enhancer. *Nature* 1–5 (2020).
4. de Visser, J. A. G. M. & Krug, J. Empirical fitness landscapes and the predictability of evolution. *Nat. Rev. Genet.* **15**, 480–490 (2014).
5. Kondrashov, D. A. & Kondrashov, F. A. Topological features of rugged fitness landscapes in sequence space. *Trends in Genetics* **31**, 24–33 (2015).
6. de Visser, J. A. G. M., Elena, S. F., Fragata, I. & Matuszewski, S. The utility of fitness landscapes and big data for predicting evolution. *Heredity (Edinb)* **121**, 401–405 (2018).
7. Weirauch, M. T. & Hughes, T. R. Conserved expression without conserved regulatory sequence: the more things change, the more they stay the same. *Trends Genet.* **26**, 66–74 (2010).
8. Orr, H. A. The genetic theory of adaptation: a brief history. *Nature Reviews Genetics* **6**, 119–127 (2005).
9. Weinreich, D. M., Lan, Y., Wylie, C. S. & Heckendorn, R. B. Should evolutionary geneticists worry about higher-order epistasis? *Current Opinion in Genetics & Development* **23**, 700–707 (2013).
10. Venkataram, S. *et al.* Development of a Comprehensive Genotype-to-Fitness Map of Adaptation-Driving Mutations in Yeast. *Cell* **166**, 1585-1596.e22 (2016).
11. Keren, L. *et al.* Massively Parallel Interrogation of the Effects of Gene Expression Levels on Fitness. *Cell* **166**, 1282-1294.e18 (2016).
12. Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature* **533**, 397–401 (2016).
13. Ogden, P. J., Kelsic, E. D., Sinai, S. & Church, G. M. Comprehensive AAV capsid fitness landscape reveals a viral gene and enables machine-guided design. *Science* **366**, 1139–1143 (2019).
14. Pitt, J. N. & Ferré-D’Amaré, A. R. Rapid construction of empirical RNA fitness landscapes. *Science* **330**, 376–379 (2010).
15. Shultzaberger, R. K., Malashock, D. S., Kirsch, J. F. & Eisen, M. B. The Fitness Landscapes of cis-Acting Binding Sites in Different Promoter and Environmental Contexts. *PLOS Genetics* **6**, e1001042 (2010).
16. Mustonen, V., Kinney, J., Callan, C. G. & Lässig, M. Energy-dependent fitness: a quantitative model for the evolution of yeast transcription factor binding sites. *Proc. Natl. Acad. Sci. U.S.A.* **105**, 12376–12381 (2008).
17. Hartl, D. L. What Can We Learn From Fitness Landscapes? *Curr Opin Microbiol* **0**, 51–57 (2014).

18. Otwinowski, J. & Nemenman, I. Genotype to phenotype mapping and the fitness landscape of the *E. coli* lac promoter. *PLoS ONE* **8**, e61570 (2013).
19. Sinai, S. & Kelsic, E. D. A primer on model-guided exploration of fitness landscapes for biological sequence design. *arXiv:2010.10614 [cs, q-bio]* (2020).
20. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods* **12**, 931–934 (2015).
21. Avsec, Ž. *et al.* Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat. Genet.* **53**, 354–366 (2021).
22. Shrikumar, A., Greenside, P. & Kundaje, A. Learning Important Features Through Propagating Activation Differences. *arXiv [cs.CV]* (2017).
23. Avsec, Ž. *et al.* Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* **18**, 1196–1203 (2021).
24. Fragata, I., Blanckaert, A., Louro, M. A. D., Liberles, D. A. & Bank, C. Evolution in the light of fitness landscape theory. *Trends in Ecology & Evolution* **34**, 69–82 (2019).
25. Payne, J. L. & Wagner, A. The causes of evolvability and their evolution. *Nature Reviews Genetics* **20**, 24–38 (2019).
26. de Boer, C. G. *et al.* Deciphering eukaryotic gene-regulatory logic with 100 million random promoters. *Nat. Biotechnol.* **38**, 56–65 (2020).
27. Crocker, J. *et al.* Low affinity binding site clusters confer hox specificity and regulatory robustness. *Cell* **160**, 191–203 (2015).
28. Habib, N., Wapinski, I., Margalit, H., Regev, A. & Friedman, N. A functional selection model explains evolutionary robustness despite plasticity in regulatory networks. *Mol. Syst. Biol.* **8**, 619 (2012).
29. Gillespie, J. H. Molecular Evolution Over the Mutational Landscape. *Evolution* **38**, 1116–1129 (1984).
30. Jerison, E. R. & Desai, M. M. Genomic investigations of evolutionary dynamics and epistasis in microbial evolution experiments. *Curr. Opin. Genet. Dev.* **35**, 33–39 (2015).
31. Sæther, B.-E. & Engen, S. The concept of fitness in fluctuating environments. *Trends Ecol. Evol.* **30**, 273–281 (2015).
32. Vaswani, A. *et al.* Attention is All you Need. in *Advances in Neural Information Processing Systems 30* (eds. Guyon, I. *et al.*) 5998–6008 (Curran Associates, Inc., 2017).
33. Weirauch, M. T. *et al.* Evaluation of methods for modeling transcription factor sequence specificity. *Nat. Biotechnol.* **31**, 126–134 (2013).
34. Yang, N. & Bielawski, N. Statistical methods for detecting molecular adaptation. *Trends Ecol. Evol. (Amst.)* **15**, 496–503 (2000).
35. Moses, A. M. Statistical tests for natural selection on regulatory regions based on the strength of transcription factor binding sites. *BMC Evolutionary Biology* **9**, 286 (2009).
36. Rifkin, S. A., Houle, D., Kim, J. & White, K. P. A mutation accumulation assay reveals a broad capacity for rapid evolution of gene expression. *Nature* **438**, 220–223 (2005).

37. Peter, J. *et al.* Genome evolution across 1,011 *Saccharomyces cerevisiae* isolates. *Nature* **556**, 339–344 (2018).
38. Erb, I. & van Nimwegen, E. Transcription factor binding site positioning in yeast: proximal promoter motifs characterize TATA-less promoters. *PLoS One* **6**, e24279 (2011).
39. Gilad, Y., Oshlack, A. & Rifkin, S. A. Natural selection on gene expression. *Trends Genet.* **22**, 456–461 (2006).
40. Alhusaini, N. & Collier, J. The deadenylase components Not2p, Not3p, and Not5p promote mRNA decapping. *RNA* **22**, 709–721 (2016).
41. Yang, J.-R., Maclean, C. J., Park, C., Zhao, H. & Zhang, J. Intra and Interspecific Variations of Gene Expression Levels in Yeast Are Largely Neutral: (Nei Lecture, SMCBE 2016, Gold Coast). *Mol. Biol. Evol.* **34**, 2125–2139 (2017).
42. Chen, J. *et al.* A quantitative framework for characterizing the evolutionary history of mammalian gene expression. *Genome Res.* **29**, 53–63 (2019).
43. Payne, J. L. & Wagner, A. Mechanisms of mutational robustness in transcriptional regulation. *Front. Genet.* **6**, (2015).
44. Shoval, O. *et al.* Evolutionary trade-offs, Pareto optimality, and the geometry of phenotype space. *Science* **336**, 1157–1160 (2012).
45. van Dijk, D. *et al.* Finding Archetypal Spaces Using Neural Networks. in (IEEE, 2019).
46. He, X., Duque, T. S. P. C. & Sinha, S. Evolutionary origins of transcription factor binding site clusters. *Mol Biol Evol* **29**, 1059–1070 (2012).
47. Cliften, P. F. *et al.* Surveying *Saccharomyces* genomes to identify functional elements by comparative DNA sequence analysis. *Genome Res.* **11**, 1175–1186 (2001).
48. Heinz, S., Romanoski, C. E., Benner, C. & Glass, C. K. The selection and function of cell type-specific enhancers. *Nature Reviews Molecular Cell Biology* **16**, 144–154 (2015).
49. Lehner, B. Selection to minimise noise in living systems and its implications for the evolution of gene expression. *Molecular Systems Biology* **4**, 170 (2008).
50. Metzger, B. P. H., Yuan, D. C., Gruber, J. D., Duveau, F. & Wittkopp, P. J. Selection on noise constrains variation in a eukaryotic promoter. *Nature* **521**, 344–347 (2015).
51. Kosuri, S. *et al.* Composability of regulatory sequences controlling transcription and translation in *Escherichia coli*. *Proc. Natl. Acad. Sci. U. S. A.* **110**, 14024–14029 (2013).
52. Shalem, O. *et al.* Systematic dissection of the sequence determinants of gene 3' end mediated expression control. *PLoS Genet.* **11**, e1005147 (2015).
53. Kinney, J. B., Murugan, A., Callan, C. G., Jr & Cox, E. C. Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 9158–9163 (2010).
54. Sharon, E. *et al.* Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters. *Nat. Biotechnol.* **30**, 521–530 (2012).
55. Melnikov, A. *et al.* Systematic dissection and optimization of inducible enhancers in human cells using a massively parallel reporter assay. *Nature biotechnology* **30**, 271–277 (2012).

56. Kwasnieski, J. C., Mogno, I., Myers, C. A., Corbo, J. C. & Cohen, B. A. Complex effects of nucleotide variants in a mammalian cis-regulatory element. *Proceedings of the National Academy of Sciences of the United States of America* **109**, 19498–19503 (2012).
57. Kircher, M. *et al.* Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. *Nature Communications* **10**, 3583 (2019).
58. Townsley, K. G., Brennand, K. J. & Huckins, L. M. Massively parallel techniques for cataloguing the regulome of the human brain. *Nat. Neurosci.* **23**, 1509–1521 (2020).
59. Renganaath, K. *et al.* Systematic identification of cis-regulatory variants that cause gene expression differences in a yeast cross. *Elife* **9**, (2020).
60. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat Methods* **9**, 357–359 (2012).
61. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* **33**, 831–838 (2015).
62. T, C. *et al.* Opportunities and obstacles for deep learning in biology and medicine. *J R Soc Interface* **15**, (2018).
63. Avsec, Ž. *et al.* The Kipoi repository accelerates community exchange and reuse of predictive models for genomics. *Nature Biotechnology* **37**, 592–600 (2019).
64. Quang, D. & Xie, X. FactorNet: A deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods (San Diego, Calif.)* **166**, 40–47 (2019).
65. Shrikumar, A., Greenside, P. & Kundaje, A. Reverse-complement parameter sharing improves deep learning models for genomics. *bioRxiv* 103663 (2017).
66. Morrow, A. *et al.* Convolutional Kitchen Sinks for Transcription Factor Binding Site Prediction. *arXiv:1706.00125 [q-bio]* (2017).
67. Kelley, D. R., Snoek, J. & Rinn, J. L. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* **26**, 990–999 (2016).
68. Koo, P. K., Majdandzic, A., Ploenzke, M., Anand, P. & Paul, S. B. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLoS Comput. Biol.* **17**, e1008925 (2021).
69. Quang, D. & Xie, X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* **44**, e107 (2016).
70. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv [cs.LG]* (2014).
71. Abadi, M. *et al.* TensorFlow: A system for large-scale machine learning. *arXiv:1605.08695 [cs]* (2016).
72. Jouppi, N. P. *et al.* In-Datacenter Performance Analysis of a Tensor Processing Unit. *arXiv:1704.04760 [cs]* (2017).
73. Li, J., Pu, Y., Tang, J., Zou, Q. & Guo, F. DeepATT: a hybrid category attention neural network for identifying functional effects of DNA sequences. *Brief. Bioinform.* (2020) doi:10.1093/bib/bbaa159.

74. Ullah, F. & Ben-Hur, A. A self-attention model for inferring cooperativity between regulatory features. *Nucleic Acids Res.* (2021) doi:10.1093/nar/gkab349.
75. Clauwaert, J., Menschaert, G. & Waegeman, W. Explainability in transformer models for functional genomics. *Brief. Bioinform.* (2021) doi:10.1093/bib/bbab060.
76. Hinton, G. & Tieleman, T. Lecture 6.5---RmsProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning* (2012).
77. Sinai, S. *et al.* AdaLead: A simple and robust adaptive greedy search algorithm for sequence design. *arXiv:2010.02141 [cs, math, q-bio]* (2020).
78. Linder, J., Bogard, N., Rosenberg, A. B. & Seelig, G. A Generative Neural Network for Maximizing Fitness and Diversity of Synthetic DNA and Protein Sequences. *Cell Systems* **11**, 49-62.e16 (2020).
79. Brookes, David and Park, Hahnbeom and Listgarten, Jennifer. Conditioning by adaptive sampling for robust design. *Proceedings of Machine Learning Research* (2020).
80. Killoran, N., Lee, L. J., DeLong, A., Duvenaud, D. & Frey, B. J. Generating and designing DNA with deep generative models. *arXiv:1712.06148 [cs, q-bio, stat]* (2017).
81. Fortin, F.-A., Rainville, F.-M. D., Gardner, M.-A., Parizeau, M. & Gagné, C. DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research* **13**, 2171–2175 (2012).
82. Jaeger, S. A. *et al.* Conservation and regulatory associations of a wide affinity range of mouse transcription factor binding sites. *Genomics* **95**, 185–195 (2010).
83. Tanay, A. Extensive low-affinity transcriptional interactions in the yeast genome. *Genome Res.* **16**, 962–972 (2006).
84. Sniegowski, P. D. & Gerrish, P. J. Beneficial mutations and the dynamics of adaptation in asexual populations. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **365**, 1255–1263 (2010).
85. Szendro, I. G., Franke, J., Visser, J. A. G. M. de & Krug, J. Predictability of evolution depends nonmonotonically on population size. *PNAS* **110**, 571–576 (2013).
86. Orr, H. A. The Population Genetics of Adaptation: The Adaptation of Dna Sequences. *Evolution* **56**, 1317–1330 (2002).
87. Bailey, T. L. DREME: motif discovery in transcription factor ChIP-seq data. *Bioinformatics (Oxford, England)* **27**, 1653–1659 (2011).
88. de Boer, C. G. & Hughes, T. R. YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities. *Nucleic acids research* **40**, D169-79 (2012).
89. Kent, W. J. BLAT—The BLAST-Like Alignment Tool. *Genome Res.* **12**, 656–664 (2002).
90. Cherry, J. M. *et al.* Saccharomyces Genome Database: the genomics resource of budding yeast. *Nucleic Acids Res* **40**, D700–D705 (2012).
91. Smith, J. D., McManus, K. F. & Fraser, H. B. A novel test for selection on cis-regulatory elements reveals positive and negative selection acting on mammalian transcriptional enhancers. *Mol. Biol. Evol.* **30**, 2509–2518 (2013).
92. Liu, J. & Robinson-Rechavi, M. Robust inference of positive selection on regulatory sequences in the human brain. *Sci Adv* **6**, (2020).

93. Rice, D. P. & Townsend, J. P. A test for selection employing quantitative trait locus and mutation accumulation data. *Genetics* **190**, 1533–1545 (2012).
94. Denver, D. R., Morris, K., Lynch, M. & Thomas, W. K. High mutation rate and predominance of insertions in the *Caenorhabditis elegans* nuclear genome. *Nature* **430**, 679–682 (2004).
95. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* **17**, 261–272 (2020).
96. Thompson, D. A. *et al.* Evolutionary principles of modular gene regulation in yeasts. *eLife* **2**, e00603 (2013).
97. Yassour, M. *et al.* Strand-specific RNA sequencing reveals extensive regulated long antisense transcripts that are conserved across yeast species. *Genome Biology* **11**, R87 (2010).
98. Grabherr, M. G. *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology* **29**, 644–652 (2011).
99. Wu, T. D. & Watanabe, C. K. GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics* **21**, 1859–1875 (2005).
100. Wapinski, I., Pfeffer, A., Friedman, N. & Regev, A. Natural history and evolutionary principles of gene duplication in fungi. *Nature* **449**, 54–61 (2007).
101. Li, B. & Dewey, C. N. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics* **12**, 323 (2011).
102. Yates, A. D. *et al.* Ensembl 2020. *Nucleic Acids Res* **48**, D682–D688 (2020).
103. DiCarlo, J. E. *et al.* Genome engineering in *Saccharomyces cerevisiae* using CRISPR-Cas systems. *Nucleic Acids Res.* **41**, 4336–4343 (2013).
104. Fleiss, A. *et al.* Reshuffling yeast chromosomes with CRISPR/Cas9. *PLoS Genet.* **15**, e1008332 (2019).
105. Horwitz, A. A. *et al.* Efficient Multiplexed Integration of Synergistic Alleles and Metabolic Pathways in Yeasts via CRISPR-Cas. *Cell Syst* **1**, 88–96 (2015).
106. Livak, K. J. & Schmittgen, T. D. Analysis of relative gene expression data using real-time quantitative PCR and the 2(-Delta Delta C(T)) Method. *Methods* **25**, 402–408 (2001).
107. Vandesompele, J. *et al.* Accurate normalization of real-time quantitative RT-PCR data by geometric averaging of multiple internal control genes. *Genome Biol.* **3**, RESEARCH0034 (2002).
108. Teste, M.-A., Duquenne, M., François, J. M. & Parrou, J.-L. Validation of reference genes for quantitative expression analysis by real-time RT-PCR in *Saccharomyces cerevisiae*. *BMC Mol. Biol.* **10**, 99 (2009).
109. Mardones, W. *et al.* Rapid selection response to ethanol in *Saccharomyces eubayanus* emulates the domestication process under brewing conditions. *Microb. Biotechnol.* (2021) doi:10.1111/1751-7915.13803.
110. Ibstedt, S. *et al.* Concerted evolution of life stage performances signals recent selection on yeast nitrogen use. *Mol. Biol. Evol.* **32**, 153–161 (2015).

111. Rich, M. S. *et al.* Comprehensive Analysis of the SUL1 Promoter of *Saccharomyces cerevisiae*. *Genetics* **203**, 191–202 (2016).
112. Rest, J. S. *et al.* Nonlinear fitness consequences of variation in expression level of a eukaryotic gene. *Mol. Biol. Evol.* **30**, 448–456 (2013).
113. Bergen, A. C., Olsen, G. M. & Fay, J. C. Divergent MLS1 Promoters Lie on a Fitness Plateau for Gene Expression. *Mol. Biol. Evol.* **33**, 1270–1279 (2016).
114. Alstott, J., Bullmore, E. & Plenz, D. Powerlaw: a Python package for analysis of heavy-tailed distributions. *PLoS One* **9**, e85777 (2014).

## Supplementary Information

### Gigantic Parallel Reporter Assay (GPRA) experimental details

Expression measurements were performed as described in (de Boer *et al.*, 2020) (**Supplementary Fig. 1**). Briefly, a library of ~200,000,000 random 80 bp promoters was cloned in front of a YFP reporter construct within the -160:-80 region of a synthetic promoter scaffold. The promoter scaffold used throughout this study included a distal poly-T tract (5 or more Ts), and a proximal poly-A tract (5 or more As) surrounding the random 80 mers; these features are common in yeast promoters. Furthermore, the scaffold sequences were designed to exclude strong binding sites for TFs. The dual reporter plasmid used is available from AddGene (AddGene:127546) and was derived from the plasmid used by (Sharon *et al.*, 2012). This plasmid contains *URA3*, which we use as a selectable marker, a constitutive RFP (with which to control for extrinsic noise), and the YFP under variable control. Random 80 mers (and designed 80 mer libraries) were cloned into an XhoI site using Gibson assembly. The resulting libraries were transformed into *S. cerevisiae* strains lacking *URA3* using the lithium acetate method (De Boer, 2017), selecting on SD-Ura media, and ensuring that at least 100,000,000 transformants were achieved for the random high-complexity libraries and >100x coverage for designed libraries. Because this is a low copy number CEN plasmid that is segregated like a chromosome during cell division, if a yeast cell is transformed with two different promoters, subsequent cell divisions will ensure with a very high probability that the two plasmids end up in different descendant cells. For random libraries, the strain Y8205 was used, but later experiments including the designed libraries were performed in S288C::*ura3*, which is less auxotrophic. Accordingly, all cases except that in random test dataset (complex media), the models were trained on sequences assayed in one strain of yeast and tested on sequences assayed in another, likely leading to underestimation of the model's performance due to *bona fide* differences between the strains.

Yeast were grown continuously in SD-Ura over the course of two days, and kept in log phase for ~10 generations to allow for reporters to reach equilibrium prior to sorting, diluting the media by 1:4 three times during this period as necessary to keep cells in log phase (OD below 0.8). All cultures were grown in a shaker incubator, at 30°C and approximately 250 RPM. Yeast were harvested by centrifugation, washed once in ice-cold PBS, resuspended in ice-cold PBS, and kept on ice prior to and during sorting. Sorting was performed with a Moflo Astrios (Beckman Coulter) sorting in three sets of 6 bins (all equal width and adjacent) each over the course of ~8 hours, dividing the time equally for the three sets. Cells were sorted by the log ratio of RFP to YFP signal (using mCherry and GFP absorption/emission), which controls for extrinsic sources of variation that affect both reporters (*e.g.*, cell size, plasmid copy number). Once sorted,



cells were kept on ice. Sorted samples were centrifuged to pellet sorted cells, the PBS/sheath fluid aspirated, leaving ~0.5 mL remaining, then the cells resuspended in 1 mL SD-Ura, transferred to a 50mL conical tube containing 9mL media, and the sorting tube washed once with SD-Ura, and transferred to the same conical tube. This produced 18 50 mL tubes each containing ~10 mL of SD-Ura and sorted yeast cells; one per sorting bin. These were allowed to grow for 2-3 days, until all samples reached saturation. Plasmids were isolated using Qiagen spin miniprep kits, as adapted for yeast according to the manufacturer's website (<https://www.qiagen.com/ca/resources/resourcedetail?id=5b59b6b3-f11d-4215-b3f7-995a95875fc0&lang=en>). Nextera adaptors and multiplexing indices were added by PCR, indexed samples were mixed in proportion to the number of cells sorted per bin, and the resulting libraries sequenced paired-end, 76 bp each, using an Illumina Nextseq 500 and 150 cycle kits so that complete coverage of the promoter could be achieved, including overlap in the center.

The sorting bins differed slightly each time FACS was performed for a promoter library. This resulted from the inability of the cell sorter (MoFlo Astrios) to accurately preserve bin configurations on different days and between calibrations. Consequently, the 18 bins were re-assigned for each experiment, and/or laser intensities were adjusted, such that the distribution of RFP:YFP ratios were correctly positioned within the bins. Sorting bins were defined to be uniform in width (expression range) and included the vast majority (>98%) of the distribution and the entirety of the high end of expression, but leaving out the bottom tail of expression. The bottom end of expression tended to be dominated by noise and outliers with abnormally low YFP:RFP ratios. However, the sensitivity at the low end of expression increased in our experiments over time, such that model predictions (in particular for the complex media model) were squished at the low end (e.g. **Extended Data Fig. 1, 3** lower left corners).

The paired reads representing both sides of the promoter sequence were aligned using the overlapping sequence in the middle, constrained to have 40 (+/-15) bp of overlap, and discarding any reads that failed to align well within these constraints. This was not required for the designed libraries. Promoters were aligned to themselves using Bowtie2 (Langmead *et al.*, 2009) to identify clusters of related sequences, merging these clusters and taking the sequence with the most reads as the “true” promoter sequence for each cluster. The designed library reads were aligned to the promoter sequences we ordered using Bowtie2, and only perfect matches were considered in further analysis. Mean expression level for each promoter (as in the processed files) was taken as the average of the bins, weighted by the number of times the promoter was observed in each bin. For the designed libraries (that included all the high-quality test data experiments), we calculated expression for all promoters for which any reads were seen, but used only those

for which we saw at least 100 reads for the analyses described to reduce the amount of measurement error present in the data. For high-complexity random libraries, all promoters were used.

## Biochemical models

The biochemical models were created and used as described previously (de Boer *et al.*, 2020). Code is available on GitHub (<https://github.com/de-Boer-Lab/CRM2.0>). Briefly, the models are trained using the “makeThermodynamicEnhancosomeModel.py” program within the <https://github.com/de-Boer-Lab/CRM2.0/blob/master/usefulScripts/makeProgressiveBiochemicalModels.bat> script (using the “110 - eb” parameters which describe the sequence length (110) and the expected binding TF model (-eb)). Training happens in 5 stages, with each subsequent stage restoring the parameters learned in the previous step before continuing training, and optimizing the noted new parameters as well as all others that were previously learned: (1) potentiation and activity parameters are learned, after having initialized the motifs to known motifs for each TF, and TF concentrations initialized to the min  $K_d$  possible with each motif (corresponding to 50% occupancy of a perfect binding site); (2) concentration parameters are optimized; (3) motif models are optimized; (4) TF binding/activity limits are introduced and optimized; and (5) position-specific activities are introduced and optimized.

Each training round is performed with a full epoch of the training data (5 epochs total). Inference is performed using the “predictThermodynamicEnhancosomeModel.py” program. In order to get regulatory strength for a TF, the “-dotf” parameter was used, and inference run again. This parameter sets the concentration parameter of the indicated TF to 0, and then predicts expression. An example for how to use these parameters and programs to calculate regulatory complexity is included here: <https://github.com/de-Boer-Lab/CRM2.0/tree/master/usefulScripts>. For all analyses, the biochemical models using position-specific activities were used with the exception of the biochemical model-derived ECC, where the non-positional model was used, because the position-specific activity parameters we had previously found are partly dependent on the surrounding sequence context (de Boer *et al.*, 2020). The decrease in % error ( $100\% \times (1-r^2)$ ), that is, the fraction of variance unexplained relative to the biochemical model is around ~45% ( $((0.96^2 - 0.926^2) / (1 - 0.926^2))$  (positional biochemical model) for the Native test data. The biochemical models were used in sections where we required model interpretability (**Fig. 2d**), but the deep learning models were used elsewhere, since the biochemical models are slower than the deep learning models to run inference on and have lower predictive performance on the test data.

## ECC calculation details and considerations

The ECC depends on both simulated and natural variation in promoter sequences. The natural variation in promoters is not independently sampled, since promoters from closely related strains often have identical sequences. Consequently, even when there are 1,011 orthologous promoters for each gene in the 1,011 whole yeast genomes dataset, there will typically be many fewer unique promoter sequences. Meanwhile, each sequence in the simulated variation is sampled independently (to increase robustness of the estimation of the null expectation), so, here, there are often 1,011 unique promoter sequences. The simulated variation was generated by placing random mutations within the gene’s promoter consensus (the most abundant base at each position in the orthologous set), while preserving the Hamming distance distribution observed in the natural sequences (**Methods**). Despite these sets each having the same Hamming distance distribution relative to the consensus, the standard deviation (SD) calculated from N independently sampled sequences (as in the simulation) is biased towards being greater than that for N dependently sampled sequences (as in evolution), resulting in the raw ECC values being biased in favor of “conservation” as a result of a statistical bias rather than due to selection.

To demonstrate that this bias is not evolutionary in nature we calculated a “mock” ECC where both the numerator and denominator represent simulated variation. In the mock ECC, the sequences in the numerator are sampled independently and match the Hamming distance distribution of the natural variation (as in the standard ECC), but the denominator (normally the natural variation) is sampled in a way that matches *both* the Hamming distance distribution *and* the number of unique sequences at each Hamming distance (relative to the natural variation). Despite both sets of sequences being randomly sampled and having matched Hamming distance distributions, the mock ECC is slightly positively biased (**Supplementary Fig. 2a**), highlighting the need for a correction factor. Consequently, we used the median of these mock ECCs  $\left(\log_2\left(\frac{\sigma_{C_i}}{\sigma_{C_i}}\right)\right)$  as the correction factor.

While it is theoretically better to have gene-specific correction factors, *these* are much more computationally intensive to calculate and provide little benefit in practice. To generate gene-specific correction factors, we need to make many instances of simulated variation for each gene, estimate the gene-specific bias, and use it to correct the observed ECC. Doing this with 1,111 simulations for each gene showed that there was little difference in the resulting ECC values, compared to a global correction factor (**Supplementary Fig. 2b**). Given the computational intensity of this approach (which, after optimization, still takes several days to run) and low practical utility, we favored the approach with a global correction factor. We do provide the gene-specific corrected ECCs in **Supplementary Table 1**.

The substitution rate in the genome is not uniform, but we use a uniform substitution rate when calculating the ECC. To test for the impact of this choice, we re-calculated the ECC using the substitution rates observed in the 1,011 yeast genomes promoters and found that the ECCs were largely concordant (**Supplementary Fig. 2c**). Since the mutations we observe in promoters are themselves biased (having survived selection), both approaches yield similar ECC values, and it is much easier to use a uniform base substitution rate, we use the uniform substitution rate ECC throughout the study.

Finally, we note that our approach for computing the ECC assumes that the relative effects of mutations within a sequence are similar regardless of the surrounding sequence context.

## Comparison of ECC to RNA-seq expression

We examined the robustness of our finding that ECC distributions differ significantly between genes with conserved and divergent expression (by RNA-seq) to the threshold we chose to define expression conservation. To this end, we performed the Wilcoxon rank sum test analysis across a range of thresholds for each dataset. Both the *Saccharomyces* and Ascomycota results were significant ( $P < 0.05$ ) at all thresholds, and much more significant ( $p < 10^{-5}$ ) at a threshold of 10% and above (**Supplementary Fig. 3a-c**).

For mammals, we used the threshold of 25% applied in the original publication (Chen *et al.*, 2019). In addition, we performed the Wilcoxon rank sum test analysis across a range of thresholds and found that the results were similarly significant for the full range of thresholds bar one (5%, the lowest threshold; **Supplementary Fig. 3d**). The null hypothesis could not be rejected at the 5% threshold, given the smaller number of yeast gene one-to-one orthologs in mammals in both the expression conservation classes.

In principle, the ECC can be calculated across orthologous regulatory sequences from many different species (as opposed to individuals within a species, as we did here), but we advise caution if doing so. The ECC assumes that the function relating sequence to gene expression is the same across the orthologous sequences being compared. Since regulatory sequences evolve much faster than the regulators themselves (Weirauch and Hughes, 2010), this assumption is likely a reasonable approximation within a species, but as evolutionary distances increase, regulators will diverge, gradually eroding this assumption. An alternative is to use gene orthology to infer the extent of expression conservation in one species using

ECCs calculated in another species (**Extended Data Fig. 4b**). However, such relations would extend only to well-mapped orthologs.

## Benchmarking of sequence-to-expression models

We examined different neural network architectures for their ability to predict expression when trained on our data. We compared our transformer model to three model architectures from existing literature (Agarwal *et al.*, 2020) on gene expression prediction models: DeepAtt (Li *et al.*, 2020), DeepSEA (Zhou and Troyanskaya, 2015), and DanQ (Quang and Xie, 2016). (We focus here on comparison to the transformer model, as the convolutional model was not used for some of the compared tasks, such as calculation of the ECC. However, equivalent comparisons can be made with the convolutional using the code shared) Although these models differ from our own and from each other, we adopted each of the model architectures for our application to the best of our ability using the source code (<https://github.com/jiawei6636/Bioinfor-DeepATT>) from each original publication (the adopted model architecture implementation can be found on our GitHub repo at: [https://github.com/1edv/evolution/tree/master/manuscript\\_code/model/benchmarking\\_models](https://github.com/1edv/evolution/tree/master/manuscript_code/model/benchmarking_models)) for the purpose of this benchmarking analysis. The precise details of the benchmarking architectures can be found in the code, and are described below. Note, that the input and output layers (which are the same for each model) are omitted from the lists below.

### 1) DeepATT :

- Convolution (filters=256, kernel\_size=30)
- MaxPool (pool\_size = 3, strides = 3)
- Dropout (0.2 probability)
- BiDirectional LSTM (16 units)
- MultiHeadAttention
- Dropout (0.2 probability)
- Dense (16 units)
- Dense (16 units)

### 2) DeepSEA :

- Convolution (filters=320, kernel\_size=8)
- MaxPool (pool\_size = 3, strides = 3)

- Dropout (0.2 probability)
- Convolution ( filters=480, kernel\_size=8)
- MaxPool (pool\_size = 3, strides = 3)
- Dropout (0.5 probability )
- Dense (64 units)
- Dense (64 units)

3) DanQ :

- Convolution (filters=320, kernel\_size=26)
- MaxPool (pool\_size = 3, strides = 3)
- Dropout (0.2 probability)
- BiDirectional LSTM (320 units)
- Dropout (0.5 probability)
- Dense (64 units)
- Dense (64 units)

Next, we trained each of these adapted models using the same training data (in complex media) as the original convolutional and transformer model, and tested each of the model's predictive power on a set of high-quality native DNA sequences measured in our system. We found that our transformer model outperformed the other three architectures on these data (**Supplementary Fig. 4a**), as expected given that these other approaches were designed for other purposes.

We also used each of these models to calculate the ECC, finding that the resulting ECC values are highly correlated to the ECCs predicted by the transformer model (**Supplementary Fig. 4b-d**). This shows that our framework leads to equivalent biological conclusions when used with model architectures that have overall comparable predictive performance.

To rule out the possibility that the transformer model's increased performance results from learning of technical biases, we compared the transformer model's ECC to an ECC calculated using the interpretable biochemical model (de Boer *et al.*, 2020), also trained using GPR data, which, with a single convolutional layer and many fewer parameters, is presumably less able to capture technical biases. Here too, we found that the ECCs are highly similar between the two models (**Supplementary Fig. 5g**). Finally, we found that the ECC values computed using the transformer model are better at predicting expression conservation as measured by RNA-seq across the range of possible thresholds considered (**Supplementary Fig. 5h**).

## Ablation analysis of the sequence-to-expression transformer model

The transformer model was motivated by several intuitions aimed to help it leverage known aspects of *cis*-regulation(Weirauch *et al.*, 2013; Brodsky *et al.*, 2020), but which may or may not be explicitly captured. The first convolutional block with three layers, was motivated by the idea to identify sites that are important for computing the expression target, and could be analogous to a TF scanning the length of the sequence for binding sites. The first layer was aimed towards an abstract representation of first order TF-sequence interactions by operating with convolutional kernels on the sequence in the forward and reverse strands separately to generate strand-specific features (each individual kernel in the first layer can be thought of as possibly learning the motif of one TF, or a combined representation of the motifs)(Alipanahi *et al.*, 2015; Zhou and Troyanskaya, 2015; Shrikumar, Greenside and Kundaje, 2017; Quang and Xie, 2019) and we designed the width of the first convolutional layer (30 bp) to be sufficient to capture the largest TF motifs known in yeast(de Boer and Hughes, 2012); the second was aimed towards capturing interactions between strands, by using a 2D convolution (implemented using the *tf.keras.layers.Conv2D* layer, and convolving along the sequence dimension) on the combined features from the individual strands; and the third layer was aimed towards capturing higher order interactions, such as TF-TF cooperativity. We zero-pad the convolution blocks to allow the convolutional filters to detect motif instances near the edges of the input sequence. The second block was motivated by an analogy to combining the biochemical activities of multiple bound TFs and accounting for their positional activities. Its transformer-encoder with a multi-head self-attention module(Vaswani *et al.*, 2017) could capture relations between features extracted by the convolutional block at different positions in the sequence, by attending to them simultaneously using a scaled dot product attention function. This could be analogous to the model learning ‘where to look’ within the sequence. Then, a bidirectional Long Short-Term Memory (LSTM) layer in this block was motivated by the idea of capturing long range interactions between the sequence regions. Finally, a multi-layer perceptron block was motivated by the idea of capturing cellular operations that occur after TFs are recruited to the promoter sequence, by pooling all the features extracted from the sequence through the previous layers and learning a scaling function that transforms these abstract feature representations of biomolecular interactions into an expression estimate. While these were our motivations in architecting the model, because our focus was predictive ability and not interpretability of regulatory mechanisms, we do not know if the model in fact captured these relations in this way.

In order to determine whether any of the transformer model’s layers were superfluous, we conducted an ablation study. For each ablation experiment, we initialized a new model from scratch after removing the

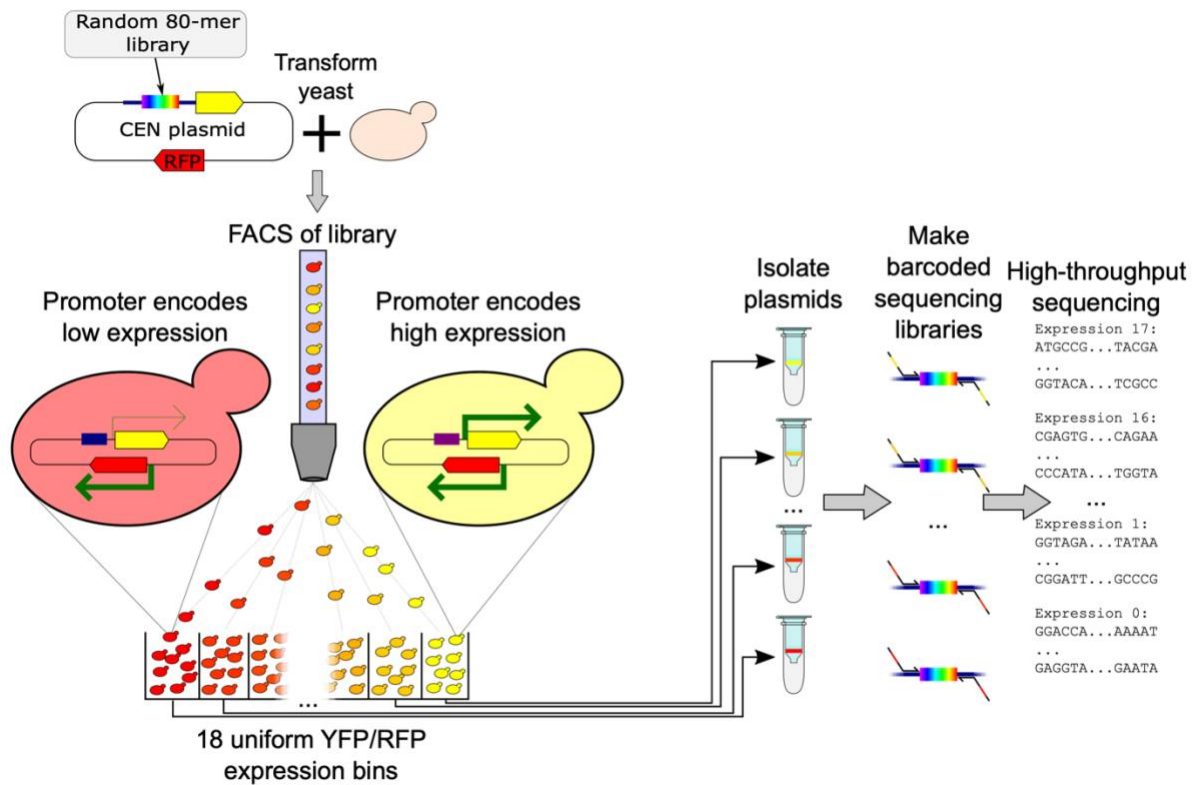
ablated layer individually from the original transformer model architecture, while retaining every other component of the original transformer model. Then, we trained this new model using the same training data (in complex media) as the original transformer model, and tested the resulting models on the high-quality random DNA test data. We found that each layer has non-trivial individual contributions to our predictions, with the full model performing better than any of the ablated models (**Supplementary Fig. 6**).

## Expression distribution at the robustness cleft and the malleable archetype

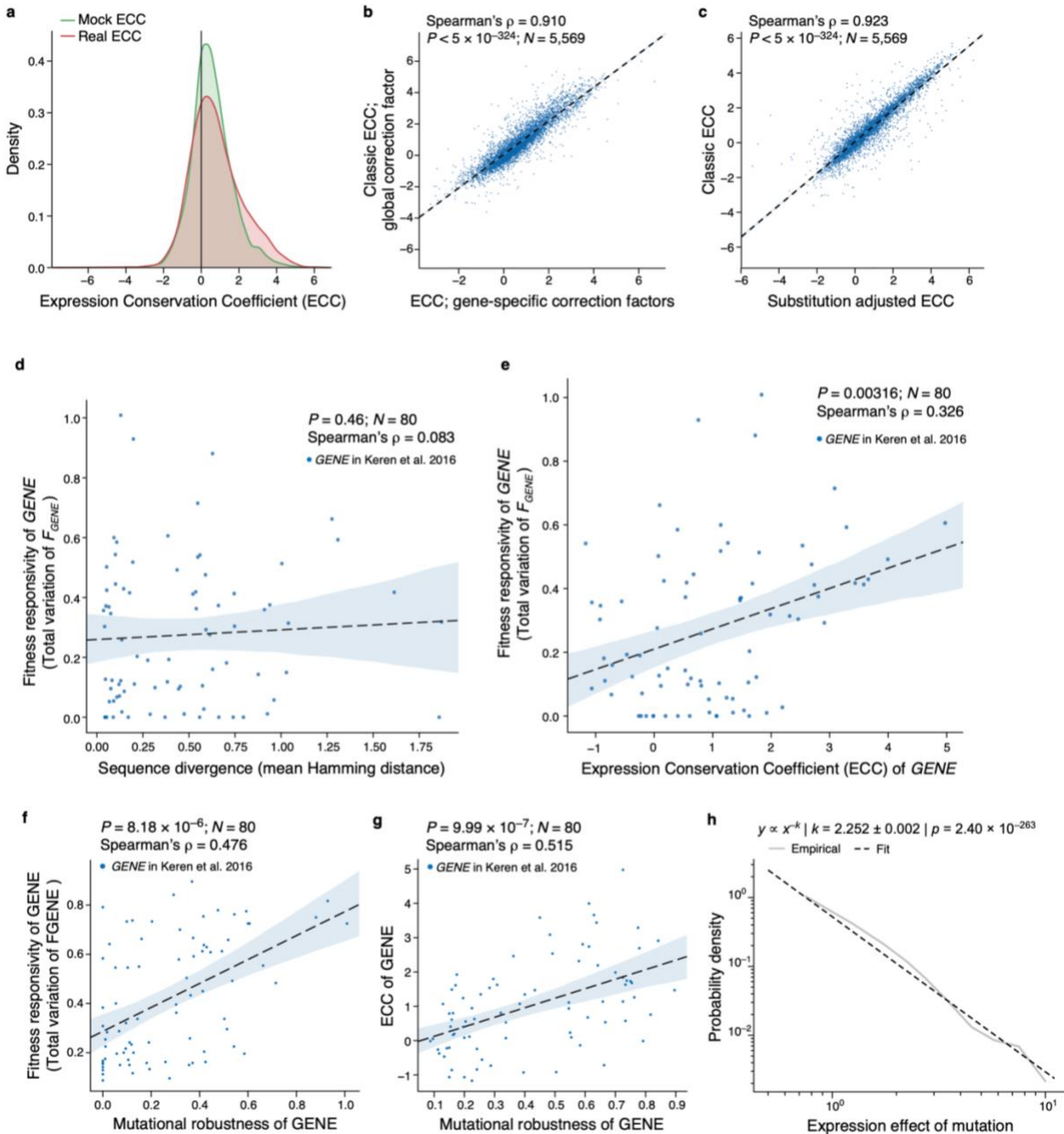
While our observation that sequences with intermediate expression levels are more likely to be near the malleable archetype ( $A_{\text{malleable}}$ ) and depleted near the robustness cleft (**Fig. 4d**), could in theory result from a saturation artifact of our reporter construct, our ratiometric sorting strategy allowed us to detect saturation and none was observed. Instead, the robustness cleft could reflect sequences at the stable extremes of one or more activation steps of gene expression (e.g. near 100% or 0% nucleosome occupied), while the malleable archetype could reflect instability around the inflection points.



## Supplementary Figures

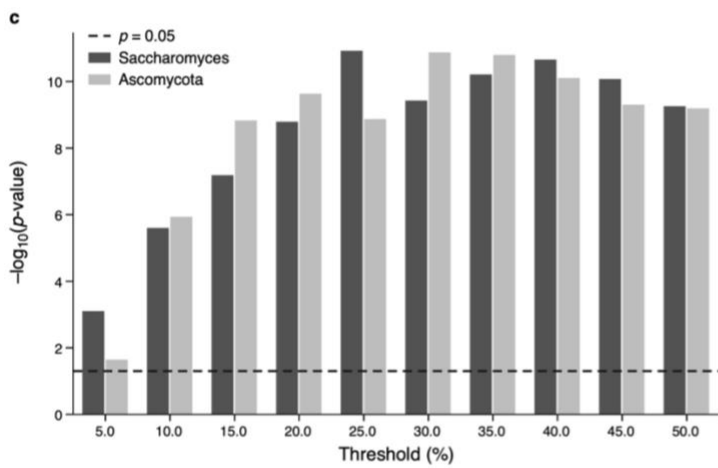
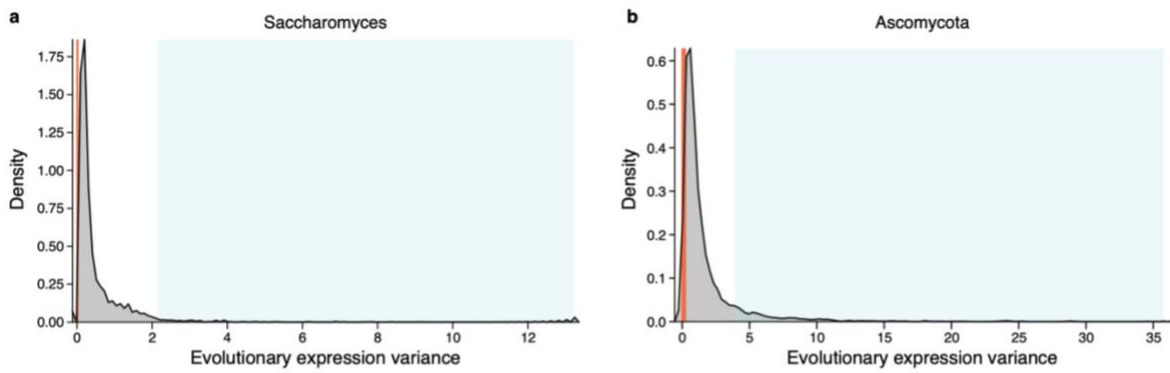


**Supplementary Fig. 1 | GPR experiment overview.** Yeast are transformed with a library of random 80 bp sequences driving YFP expression, the cells recovered and selected for successful transformants, and grown in the target media in log phase. Yeast are then sorted by the ratio of YFP to RFP into 18 different uniform expression bins. Yeast are then recovered in selection media (SD-Ura), plasmids isolated, sequencing libraries created, and the promoters in each expression bin sequenced with high-throughput sequencing.



**Supplementary Fig. 2** | **a**, Comparison of raw ECC distributions for natural variation (red) and matched simulated variation (green, “mock ECC”). Both are biased towards having an ECC above 0. **b**, Comparison of ECCs with global correction (x axis) and gene-specific correction factors (y axis). **c**, ECC with uniform substitutions (y axis) is highly correlated to the ECC computed using the observed substitution rate (x axis). **d**, **e**, Fitness responsiveness is not associated with simple sequence diversity, but is associated with ECC. Fitness responsiveness (y axes) and mean Hamming distance (**d**, x axis) or ECC (**e**, x axis) for each of 80 genes (points). **f**, **g**, Genes whose expression changes have stronger effects on organismal fitness have mutationally robust regulatory sequences. Mutational robustness (x axes) and fitness responsiveness (**f**, y axis) or ECC (**g**; y axis) for each of 80 genes (points) for which the expression-to-fitness curves were quantified (Keren *et al.*, 2016). (**b-g**) Spearman’s  $\rho$  and associated two-tailed p-values are shown. The light blue error bands represent the respective 95% confidence intervals. **h**, Mutational effects

follow a power law distribution. Probability density ( $y$  axis) and expression effect of mutation (magnitude) ( $x$  axis) plotted on log-log axes (solid line) alongside the goodness of fit (dash line) of the power law distribution.

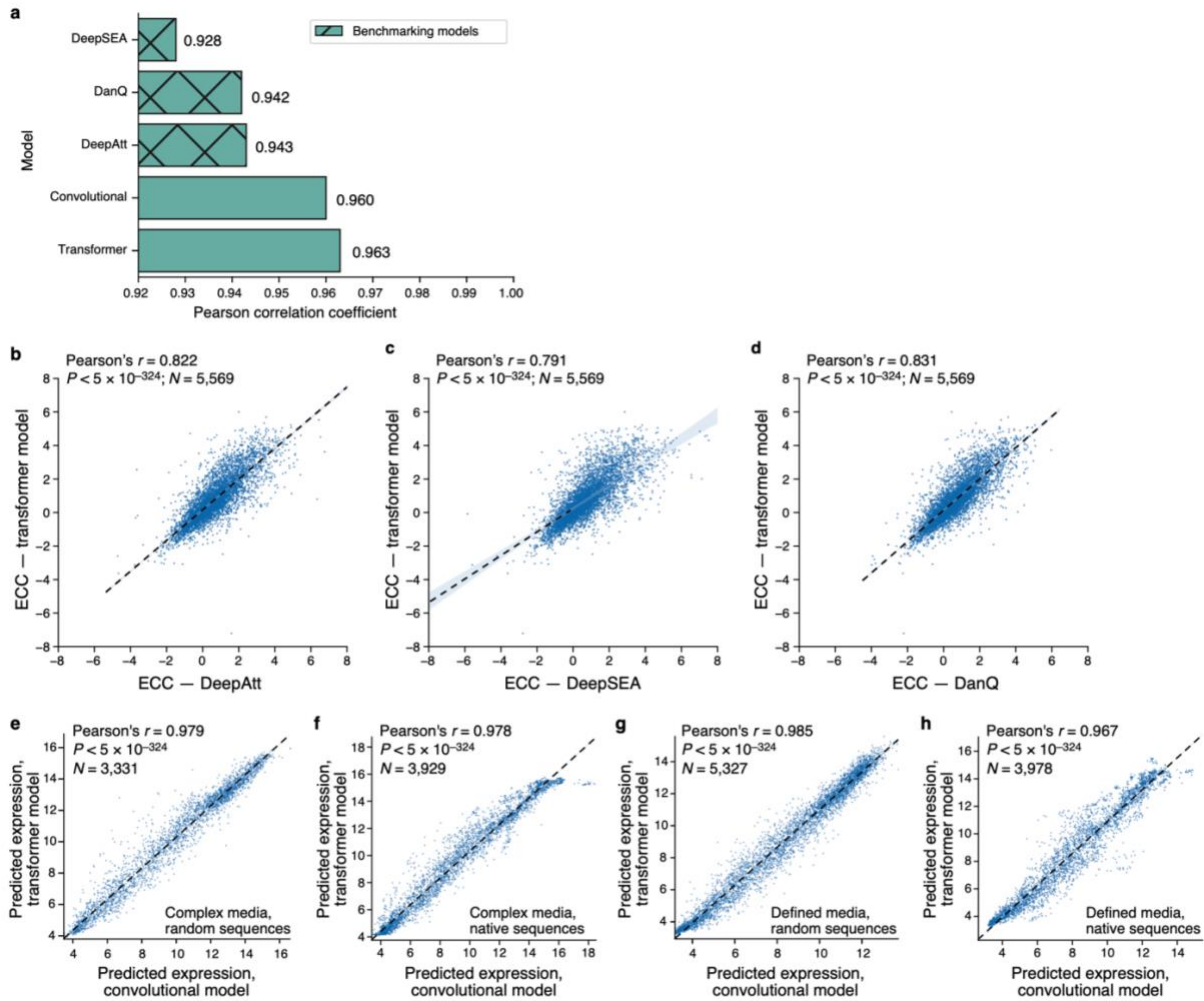


**d**

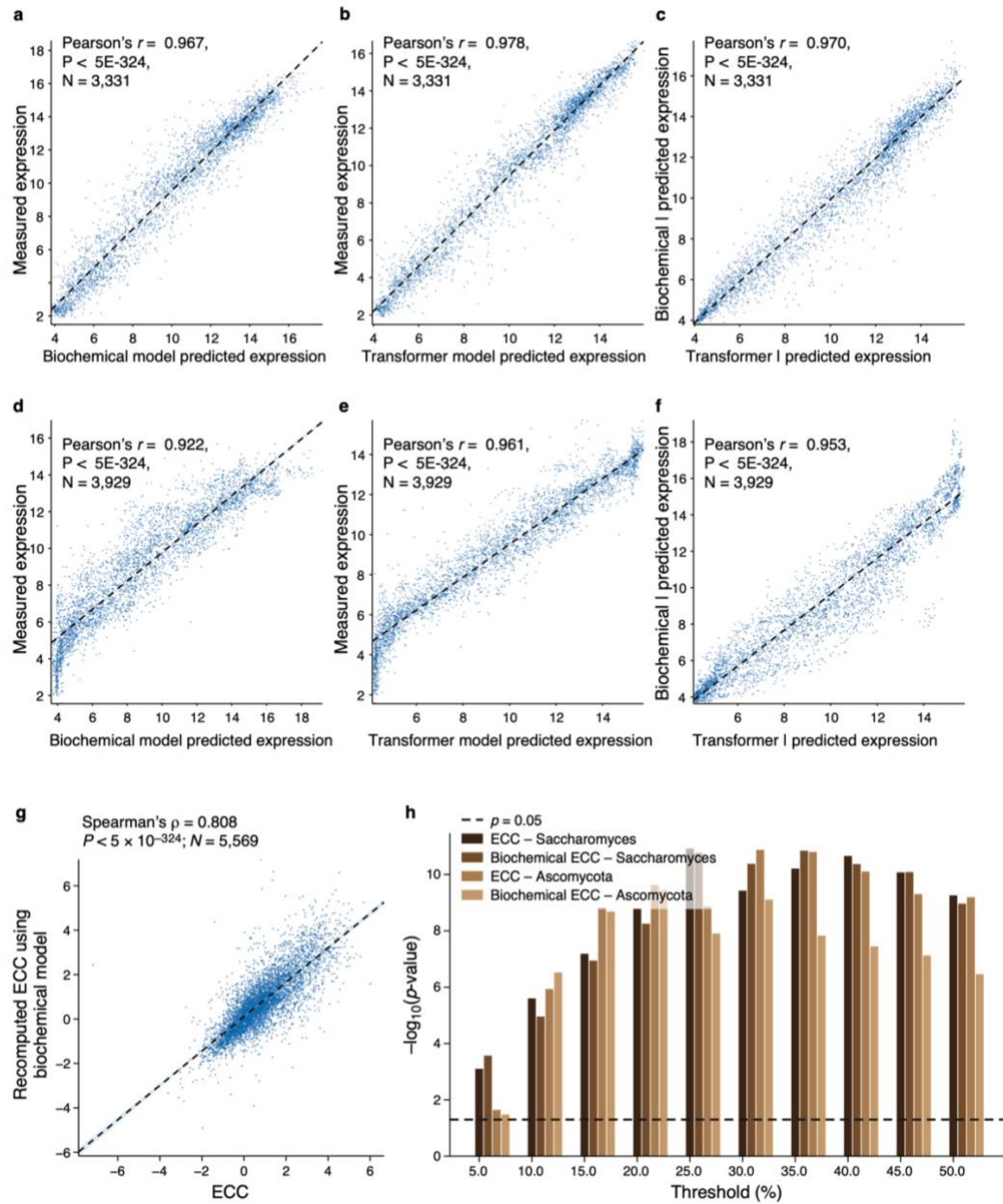
Threshold (%)	<i>n. genes (conserved)</i>	<i>n. genes (not-conserved)</i>	Mammalian p-value
5.0	144	75	0.132141
10.0	226	137	0.001095
15.0	291	208	0.000609
20.0	347	263	0.000897
25.0	333	288	0.000107
30.0	300	274	0.000040
35.0	261	246	0.00019
40.0	208	215	0.007662
45.0	173	185	0.009174
50.0	144	155	0.001364

**Supplementary Fig. 3 | a,b**, Expression variance (by RNA-seq) for *Saccharomyces* (a) and Ascomycota (b). Green boxes: genes called as divergent; orange: genes called as conserved by the thresholds in this study (as in **Extended Data Fig. 4b**). **c**, Sensitivity of ECC enrichment significance (Wilcoxon rank sum test  $-\log_{10}(P\text{-values})$ ; y axis) to “conserved” vs. “divergent” thresholds (x axis) for Ascomycota (light gray) and *Saccharomyces* (dark gray).  $P=0.05$ : dashed line. **d**, Sensitivity of ECC

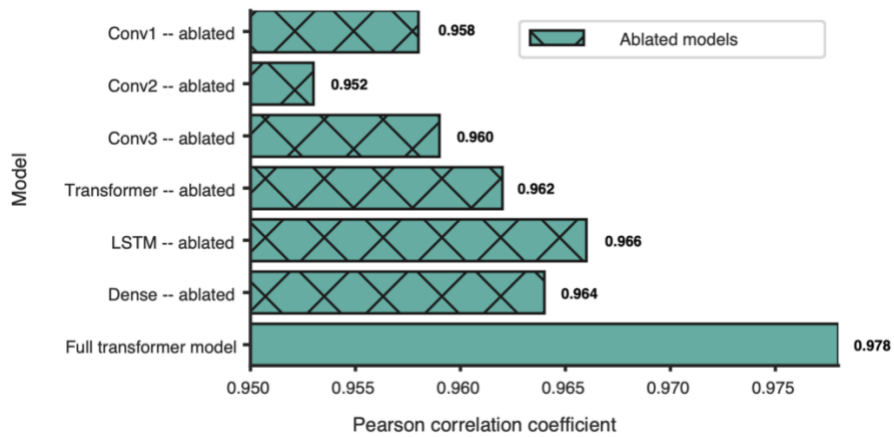
enrichment significance (Wilcoxon rank sum test, “Mammalian p-value”) to “conserved” vs. “divergent” thresholds (“Threshold (%)”) in mammals. The columns display the number of genes determined to be in each class at each threshold.



**Supplementary Fig. 4 | a**, Benchmarking of performance against existing neural network architectures. Pearson correlation coefficient between model predictions and test data (x axis) for four model (y axis). All models were trained on the same training dataset, and tested on the same set of native promoter test sequences in complex media. While all approaches performed reasonably well, the transformer model architecture used in this paper out-performed the others on the native test sequence dataset. **b-d**, Comparison of ECC calculated with our model (y axis) and with **(b)** DeepAtt, **(c)** DeepSEA and **(d)** DanQ (x axis). In each case, the ECC predictions are highly correlated between each approach and our model. (Outliers not shown for the panel **(c)** to maintain scaling and visibility; Pearson's  $r$  was computed using all of the data including outliers.). **e-h**, The convolutional and transformer models have highly correlated predictions. Predicted expression from the convolutional (x axis) and transformer (y axis) models in complex **(e-f)** and defined **(g-h)** media for random **(e-g)** and native **(f-h)** test datasets. **(b-h)** Pearson's  $r$  and associated two-tailed p-values are shown.

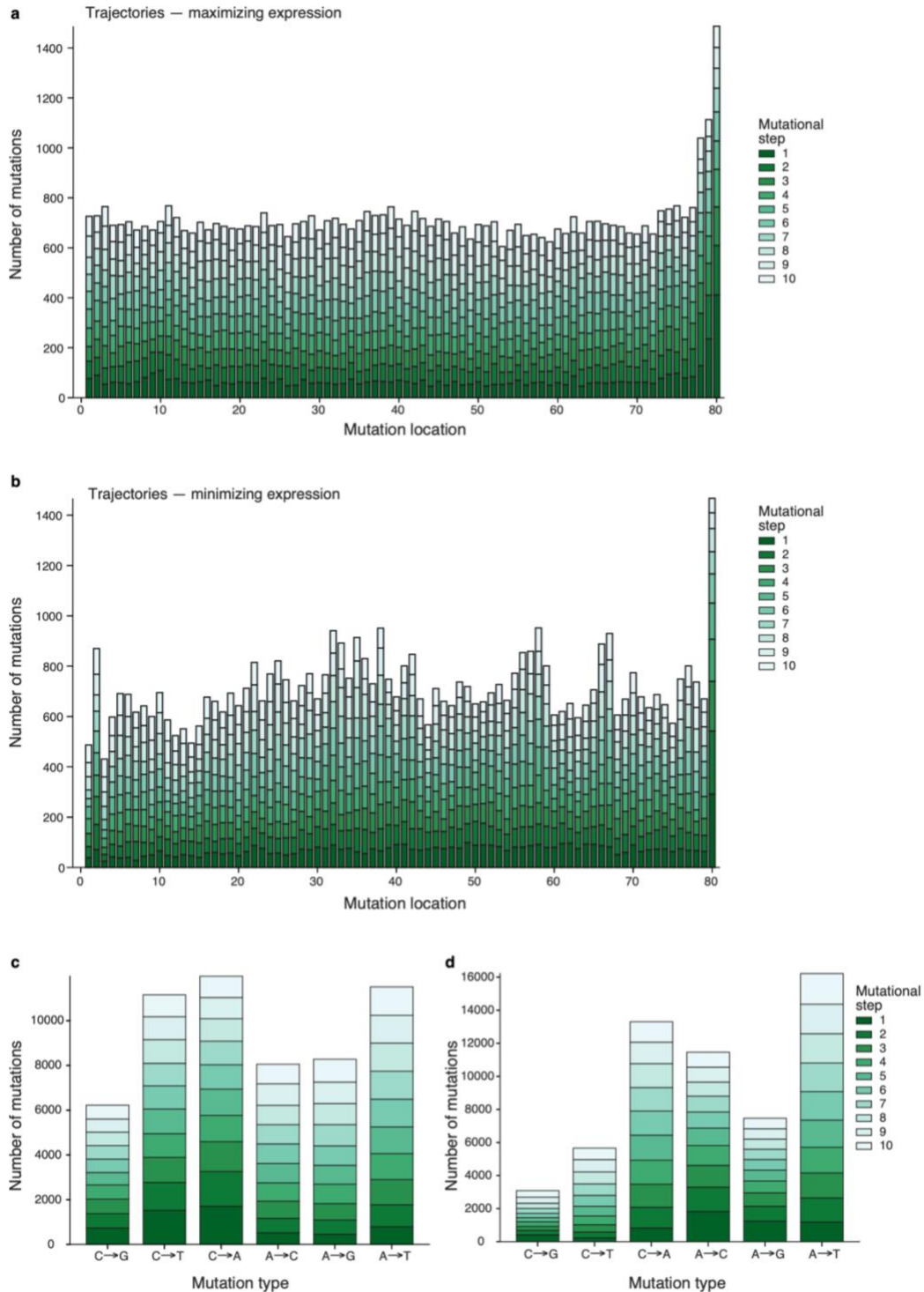


**Supplementary Fig. 5 | Comparison of the biochemical and transformer models.** Measured and predicted expression in complex media for (a-c) random test data as, and (d-f) native test data. (a,b,d,e) Measured (y-axes) and predicted (x-axes) expression, for (a,d) biochemical and (b,e) transformer models. (c,f) transformer (x-axes) and biochemical (y-axes) model predictions. (a-f) Pearson's  $r$  and associated two-tailed p-values are shown. g,h, The transformer model outperforms the biochemical model in differentiating expression conservation status. g, Comparison of ECCs calculated for each gene (points) for the transformer model (x axis) versus the biochemical model (y axis). Spearman's  $\rho$  and associated two-tailed p-values are shown. h, Significance (y axis) of rank sum statistics for how well ECCs calculated with each method separates conserved versus not conserved genes across *Saccharomyces* (dark brown) and Ascomycota (light brown).



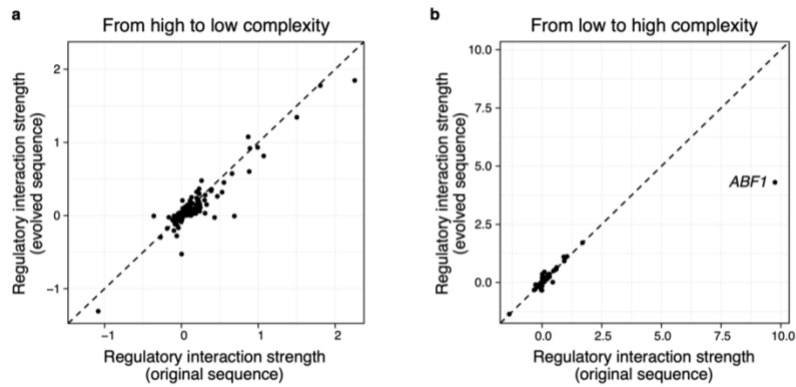
**Supplementary Fig. 6** | Each layer individually contributes to model performance. Performance ( $x$  axis, Pearson's  $r$  between the model predictions and random test data) of the transformer model variants ( $y$  axis) with each layer individually ablated, and the full transformer model (bottom). The full transformer model outperforms all other versions with any model component ablated. The two-tailed  $p$ -value corresponding to each performance metric shown is  $< 5 \cdot 10^{-234}$ .



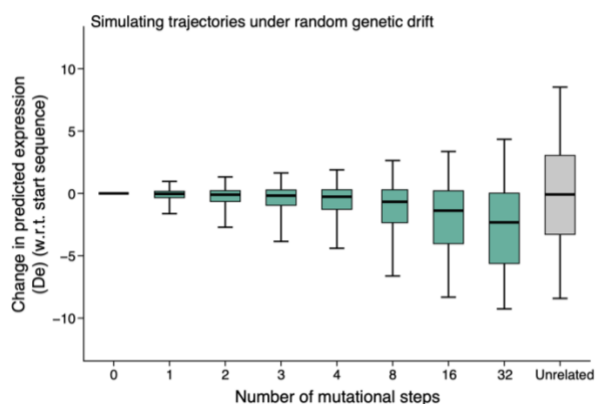


**Supplementary Fig. 7 | Sequences took diverse paths to evolve extreme expression. a-b,** The number (*y* axis) of mutations across each promoter position (*x* axis; -160 to -80 region) for trajectories under the SSWM regime starting with native promoter sequences when **(a)** maximizing or **(b)** minimizing expression in defined media using the convolutional model. Some of the observed bias to TSS-proximal mutations may be related to prior observations of proximal repressor activity bias (de Boer *et al.*, 2020). **c,d,** The number (*y* axis) of mutations of each type (*x* axis) for trajectories under the SSWM regime starting with native

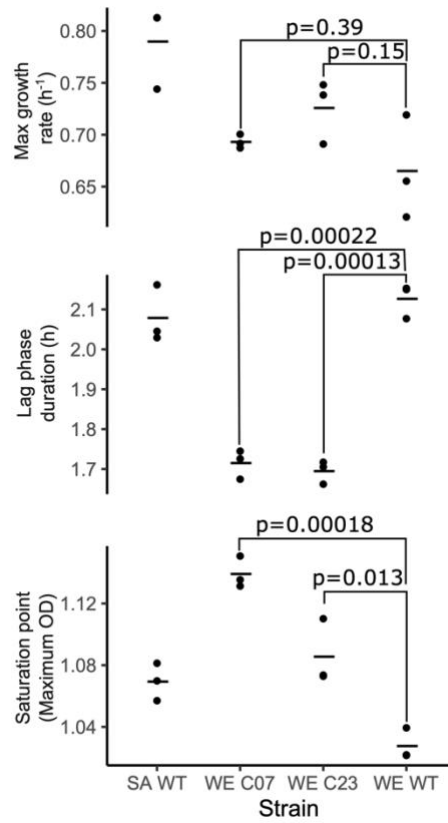
promoter sequences when **(c)** maximizing or **(d)** minimizing expression in defined media. Colors represent the mutational step (1-10).



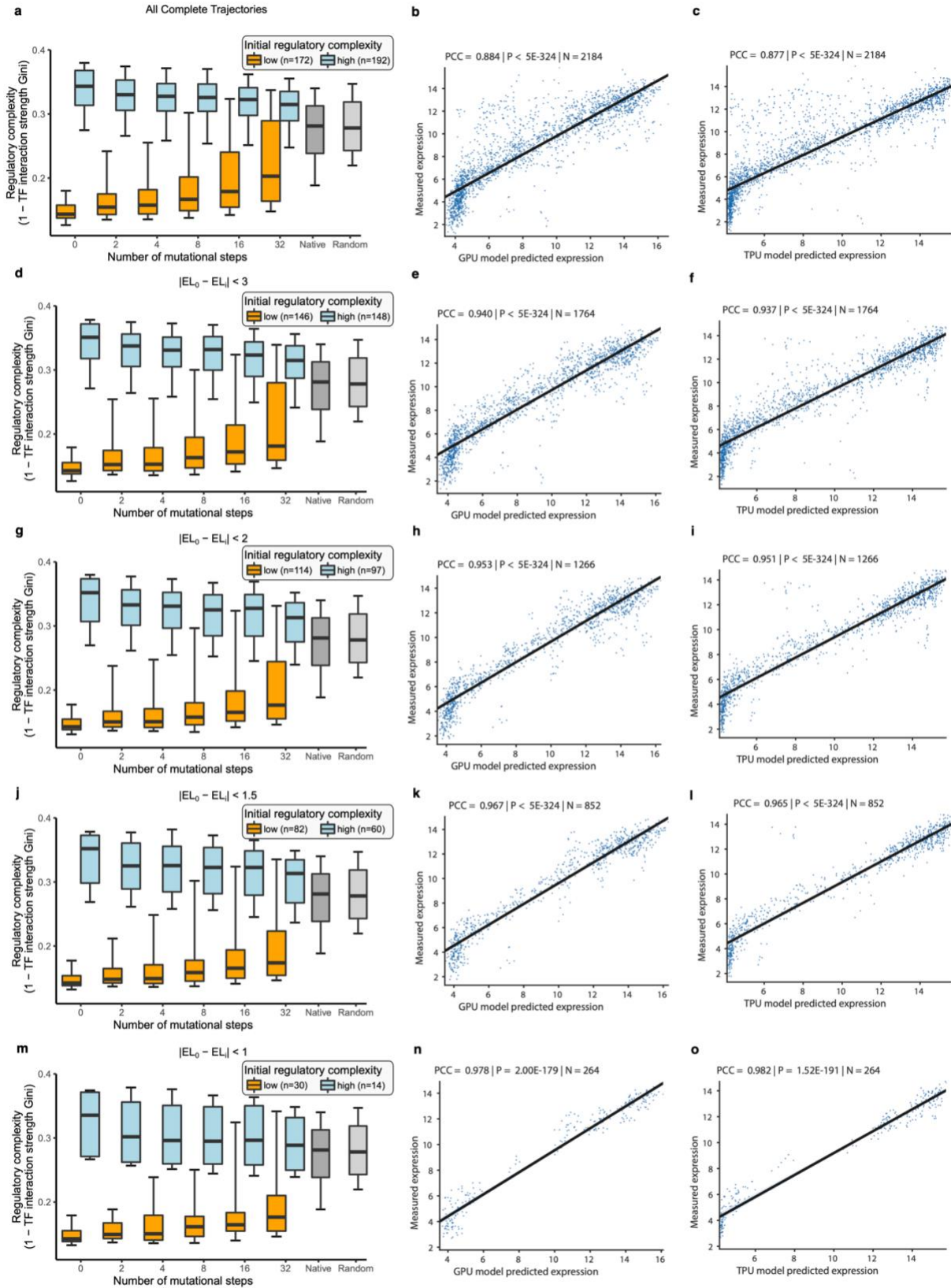
**Supplementary Fig. 8 | a,b,** Examples of regulatory complexity changes under stabilizing selection. TF regulatory interaction strengths for original ( $x$  axes) and evolved ( $y$  axes) sequences after 32 neutral (expression maintaining) mutations for each TF (points) for -160:-80 promoter regions for (a) *YDR476C*, whose regulatory complexity was high and decreased (from 0.3 to 0.25), and (b) *AIF1*, whose complexity was low (dominated by the TF Abf1p) and increased (from 0.14 to 0.21). Both have approximately the same predicted expression levels (13.7 and 14.3 respectively).



**Supplementary Fig. 9** | Predicted expression divergence under random genetic drift. Distribution of the change in predicted expression (y axis) for native yeast promoter sequences (n=5,720) at each mutational step (x axis) for trajectories simulated under random mutational drift using the transformer model. Silver bar: differences in expression between unrelated sequences. Expression decreases with increasing mutation number because the average expression of the starting set of native sequences is greater than for random DNA, and so including random mutations are more likely to decrease expression than increase it. Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentiles.

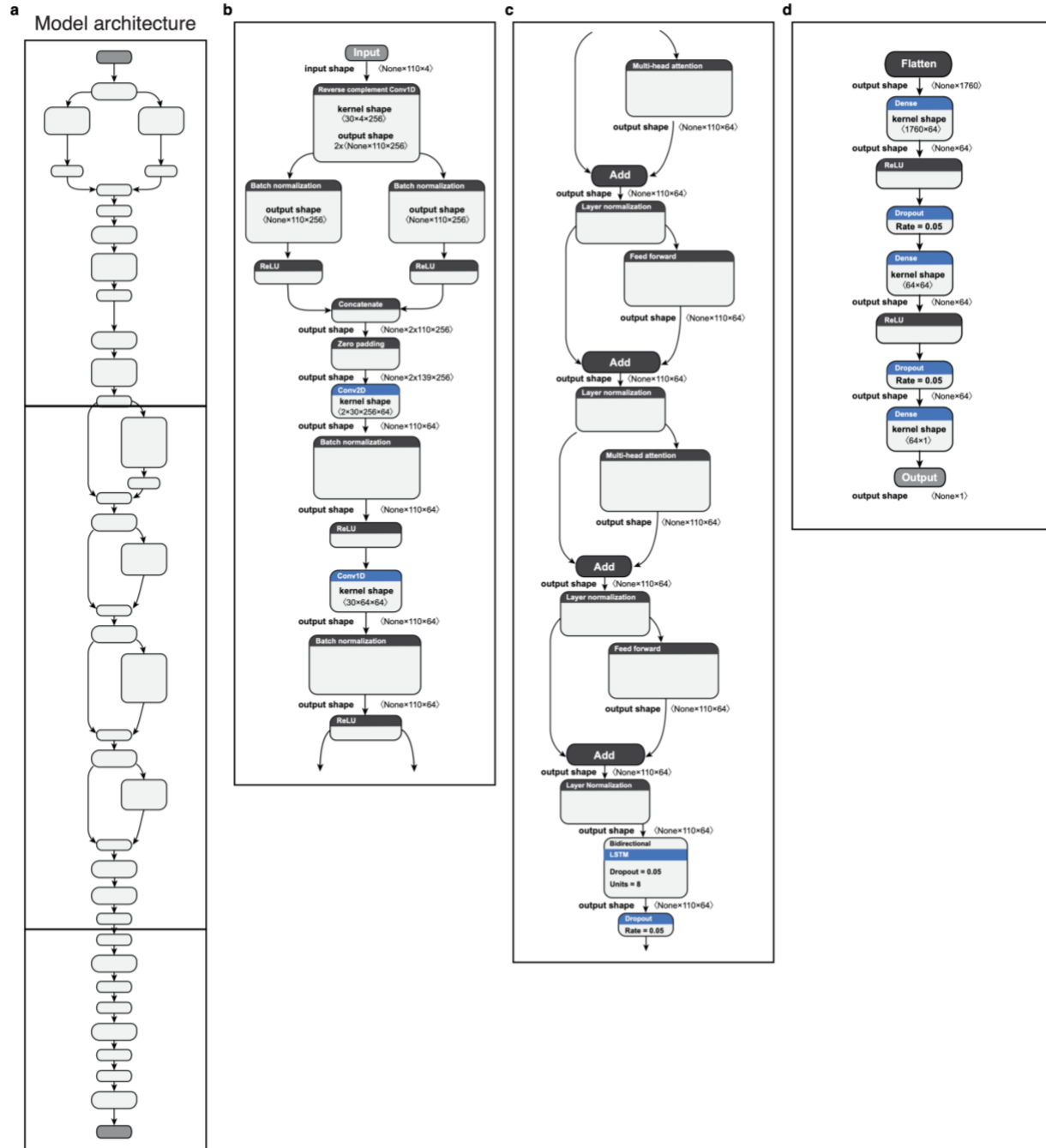


**Supplementary Fig. 10** | Growth phenotypes of *CDC36* promoter mutant strains. Maximum growth (*y* axis, top), duration of lag phase (*y* axis, middle) and saturation of growth (*y* axis, bottom) for two WT strains and two engineered strains (*x* axis). Bars: means, dots: replicate measurements. P-values: Student's *t*-test; two-sided, unpaired, equal variance.  $n=3$  replicates/strain.



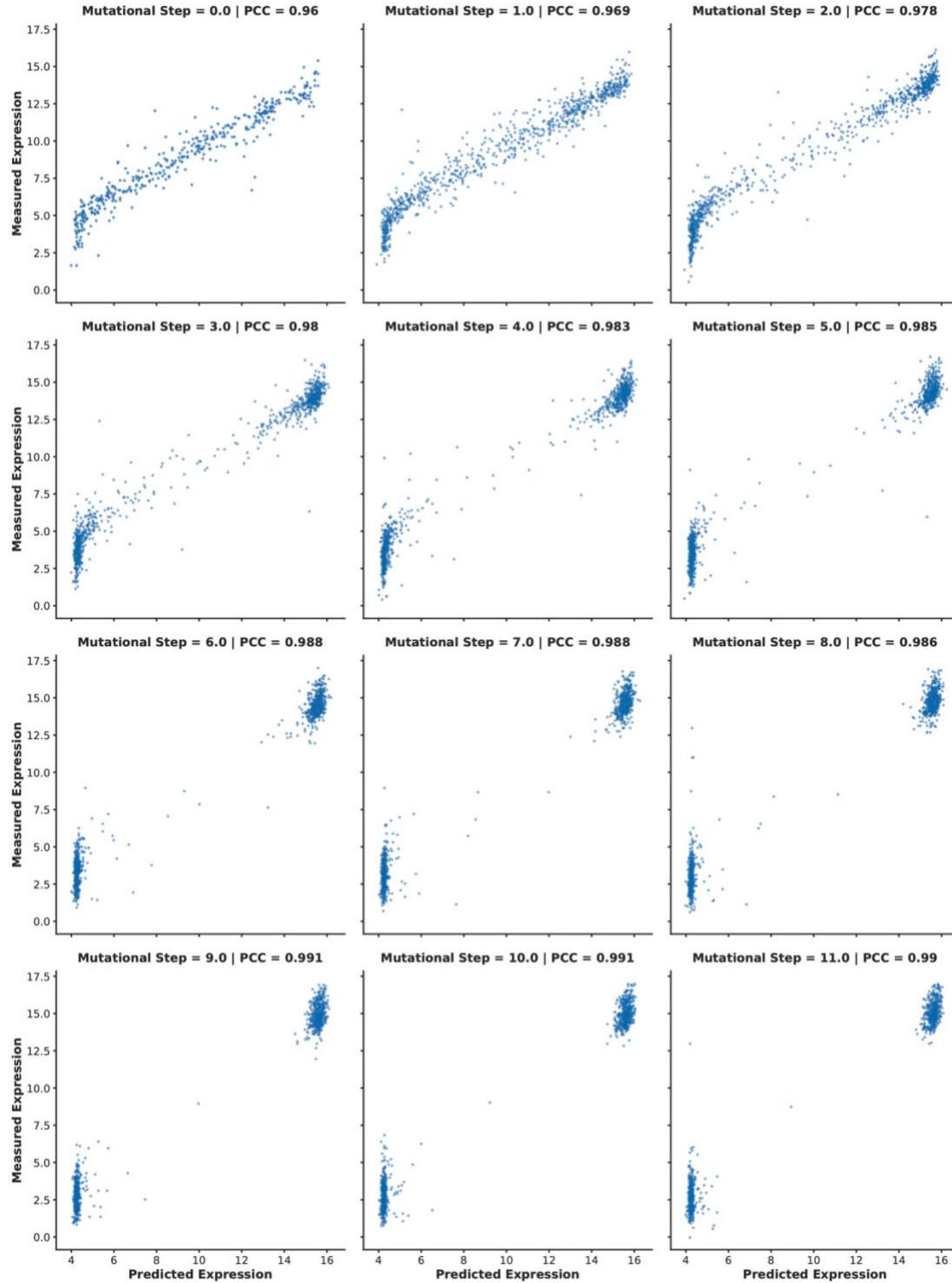
**Supplementary Figure 11: Robustness of moderation of regulatory complexity to the degree of stabilizing selection.** (a,d,g,j,m) Distributions of regulatory complexity (y-axes) for sets of sequences with initial high (light blue) and low (orange)

regulatory complexity, and evolved sequences at different mutation steps, with native and random sequences shown for reference (dark and light gray respectively). Here,  $n$  is the number of trajectories included. All evolved sequences were designed to mimic stabilizing selection by requiring that expression changes by no more than 0.5 expression units relative to the original using the GPU model. Also shown are the measured ( $y$ -axes) and model predicted ( $x$ -axes) expression levels for the convolutional (**b,e,h,k,n**) and transformer (**c,f,i,l,o**) models. Results are shown for all complete experimental trajectories (**a-c**), or when including only trajectories where no evolved sequences had measured or transformer model-predicted expression that differed from the measured expression of the original sequence by more than 3 (**d-f**), 2 (**g-i**), 1.5 (**j-l**) or 1 (**m-o**) expression units. All data are for complex media (YPD). (**a,d,g,j,m**) Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. (**b,c,e,f,h,i,k,l,n,o**) Pearson's  $r$  and associated two-tailed  $p$ -values are shown.

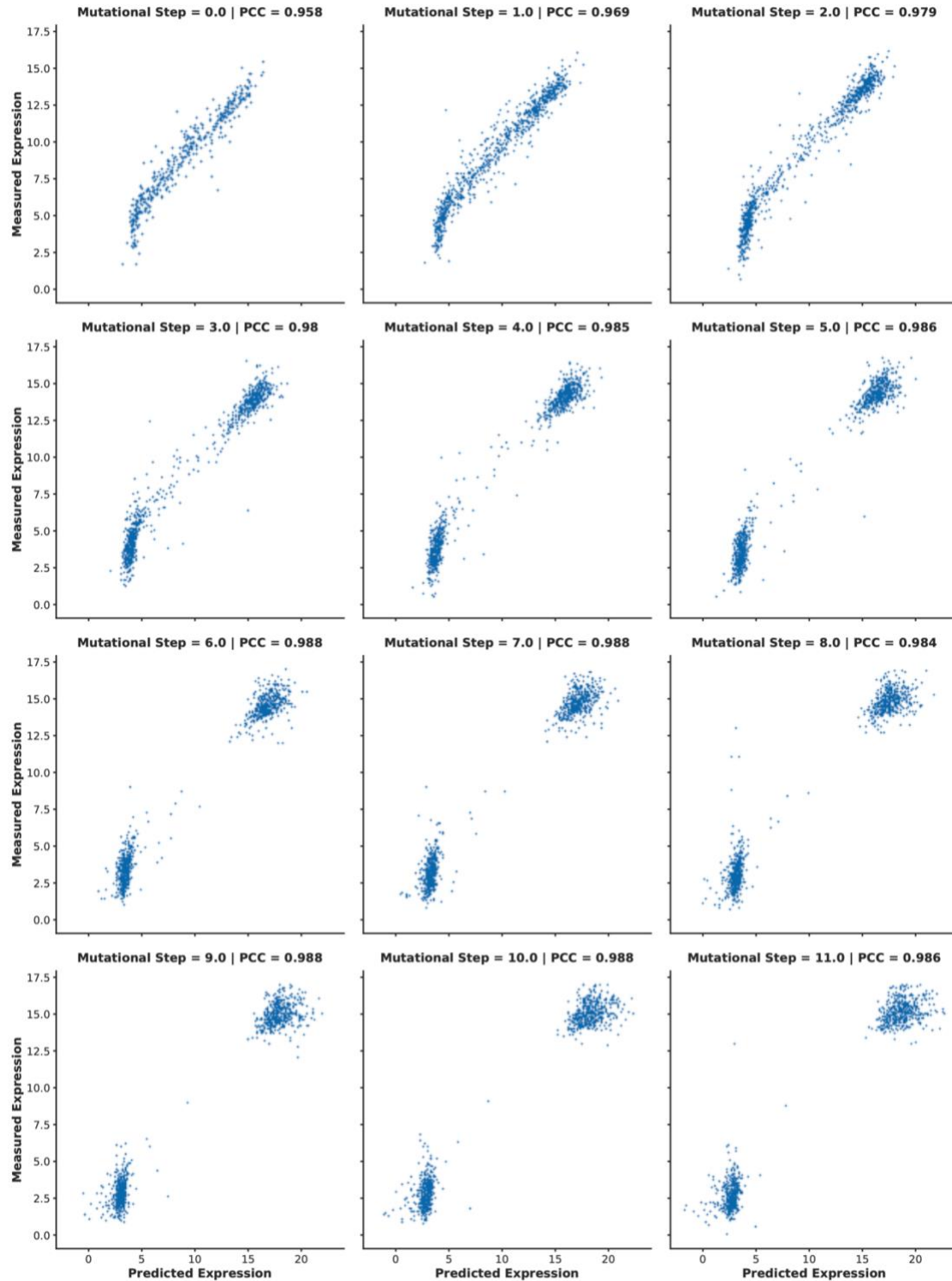


**Supplementary Fig. 12 | The deep transformer neural network architecture for the sequence-to-expression model. a,** Model architecture with three blocks (horizontal lines) and multiple layers (boxes). **b-d.** Expanded architecture (**Methods**) for the convolutional (**b**), transformer encoder (**c**) and multi-layer perceptron (**d**) blocks in our transformer model.

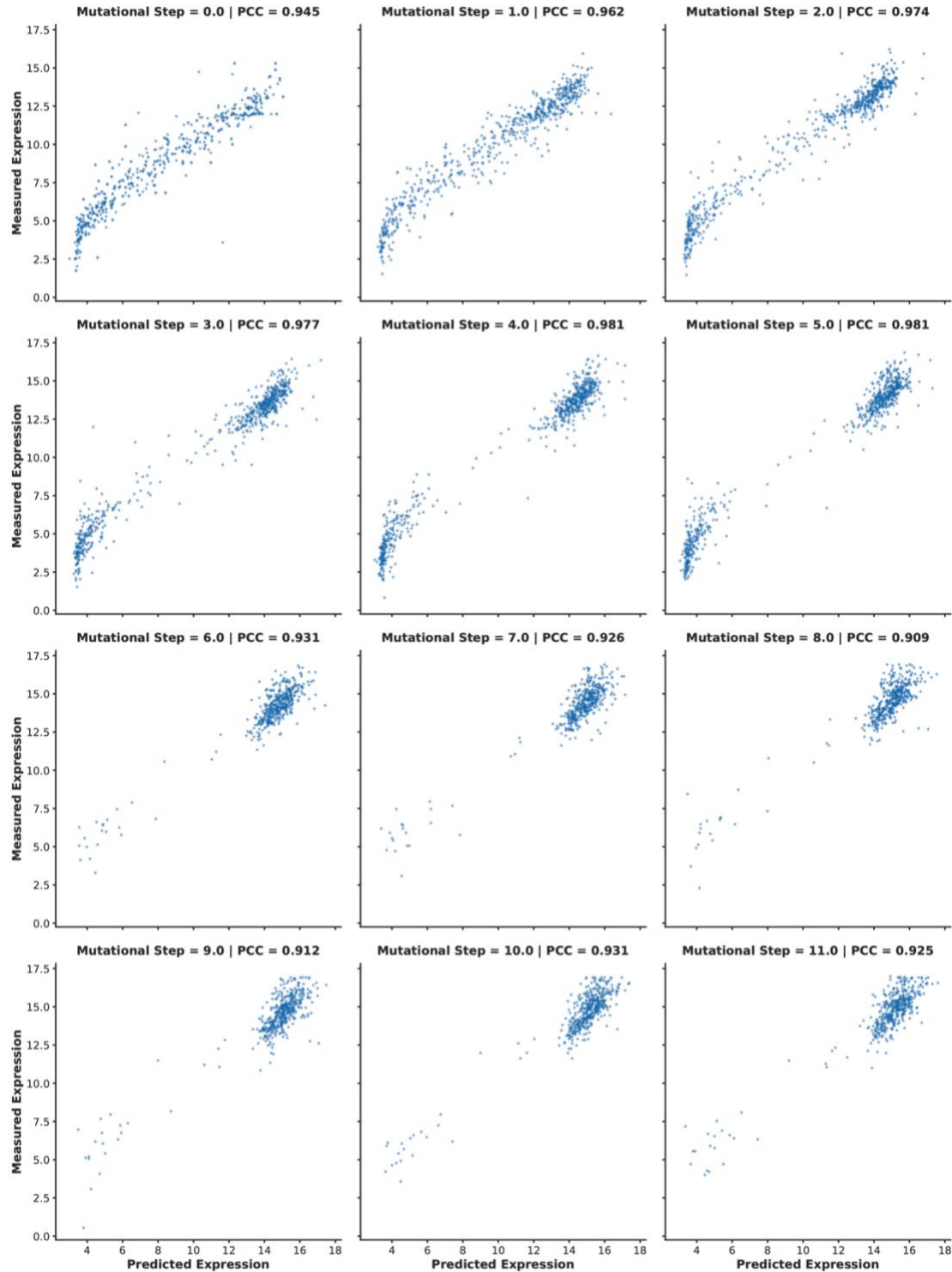




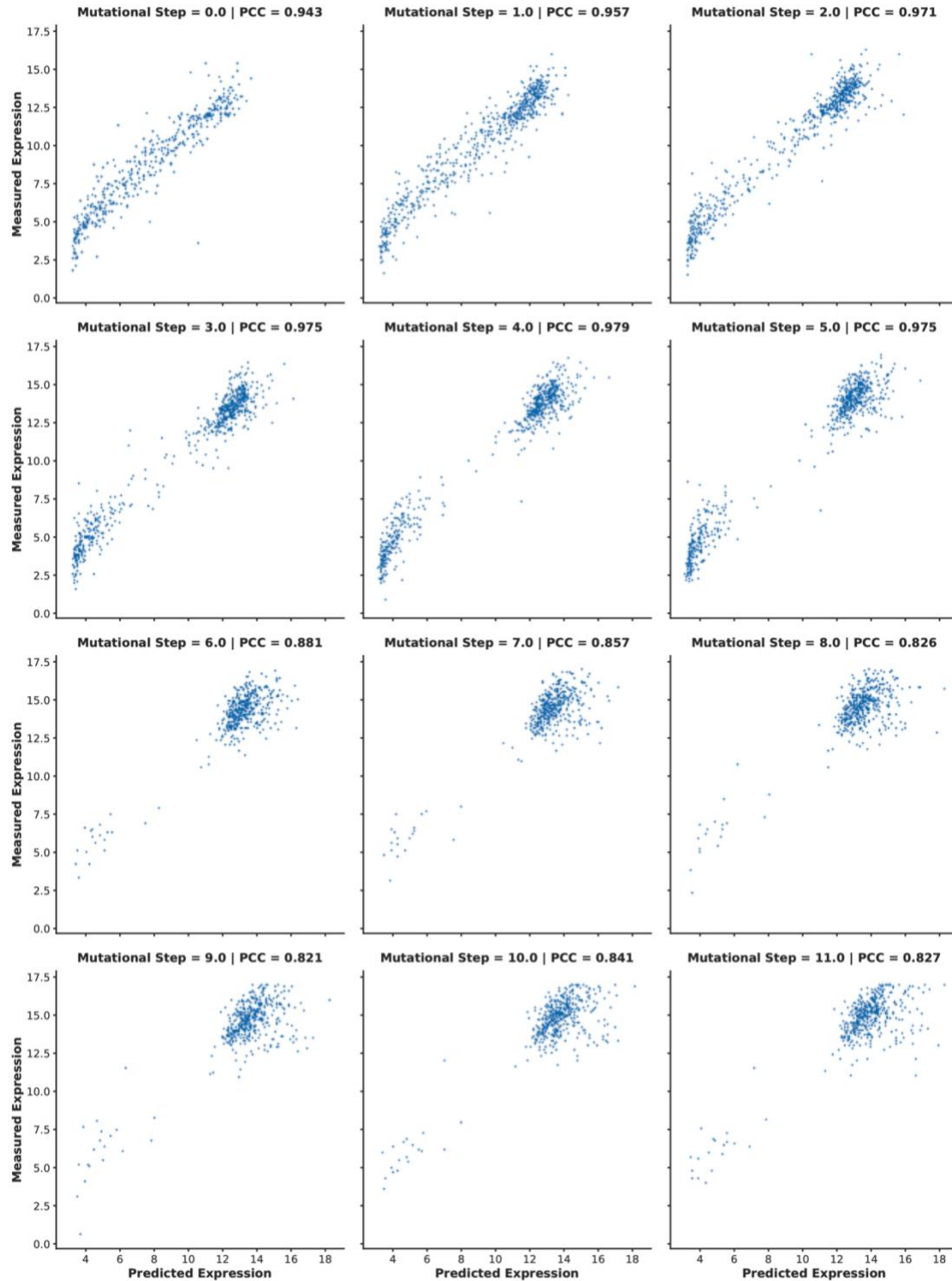
**Supplementary Fig. 13 | Comparison between predictions and measurements of expression at each mutational step in complex media.** Transformer model predicted (*x*-axes) and measured (*y*-axes) expression for each mutational step (plots) for trajectories in Fig. 2g ( $n=10,322$  sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. The two-tailed *p*-value corresponding to the performance metric shown in each panel is  $< 5 \times 10^{-234}$ .



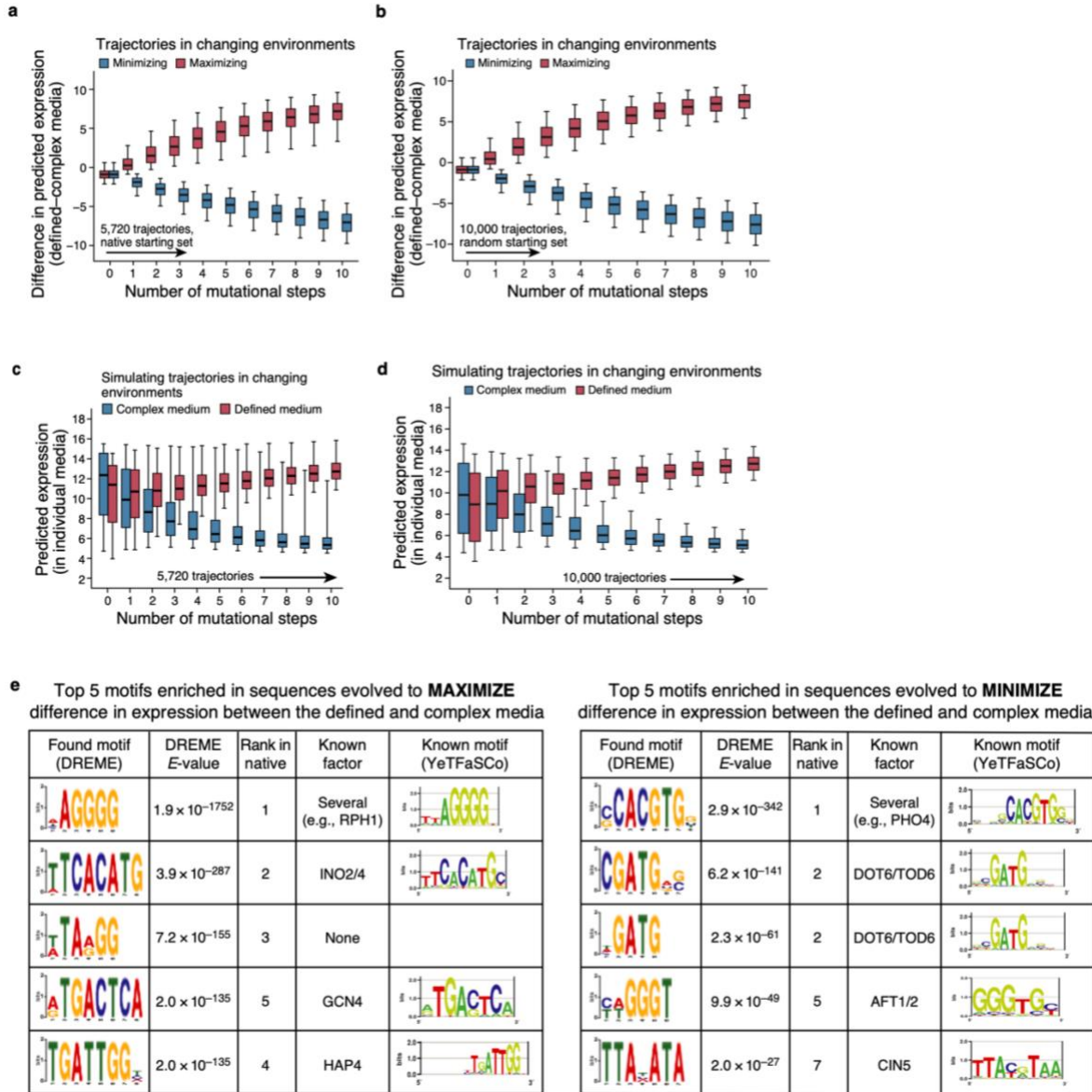
**Supplementary Fig. 14 | Comparison between predictions and measurements of expression at each mutational step in complex media.** Convolutional model predicted (*x*-axes) and measured (*y*-axes) expression for each mutational step (plots) for trajectories in Fig. 2g ( $n=10,322$  sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. The two-tailed *p*-value corresponding to the performance metric shown in each panel is  $< 5 \cdot 10^{-234}$ .



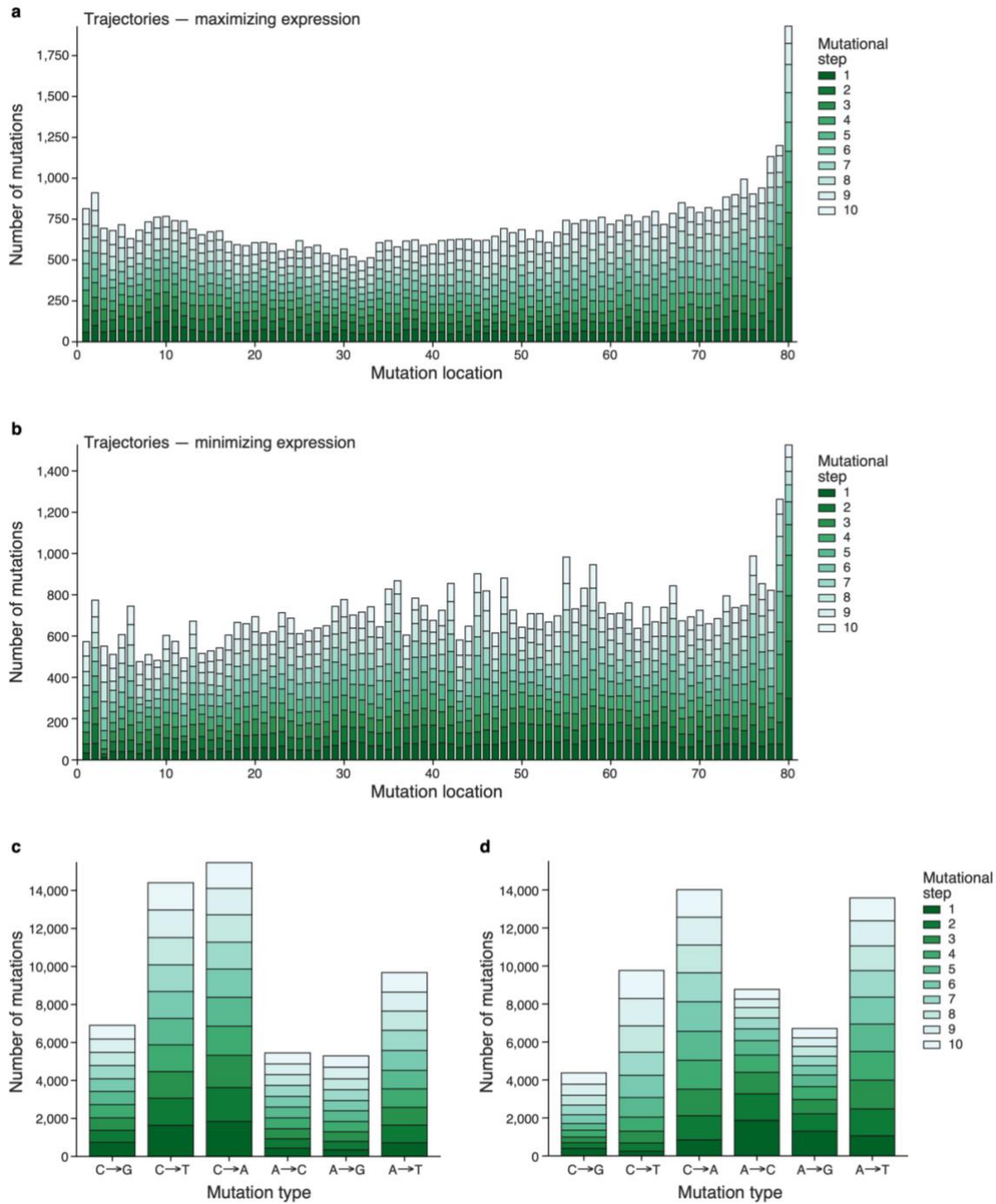
**Supplementary Fig. 15 | Comparison between predictions and measurements of expression at each mutational step in defined media.** Transformer model predicted ( $x$ -axes) and measured ( $y$ -axes) expression for each mutational step (plots) for trajectories in **Extended Data Fig. 1g** ( $n=6,304$  sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. Due to limitations in the number of sequences we could test per experiment, we only tested the decreasing expression defined media trajectory to 5 mutations. The two-tailed  $p$ -value corresponding to the performance metric shown in each panel is  $< 5 \cdot 10^{-234}$ .



**Supplementary Fig. 16 | Comparison between predictions and measurements of expression at each mutational step in defined media.** Convolutional model predicted ( $x$ -axes) and measured ( $y$ -axes) expression for each mutational step (plots) for trajectories in **Extended Data Fig. 1g** ( $n=6,304$  sequences, divided amongst plots). The Pearson's correlation coefficient (PCC) associated with each panel is also shown. Due to limitations in the number of sequences we could test per experiment, we only tested the decreasing expression defined media trajectory to 5 mutations. The two-tailed  $p$ -value corresponding to the performance metric shown in each panel is  $< 5 \cdot 10^{-234}$ .

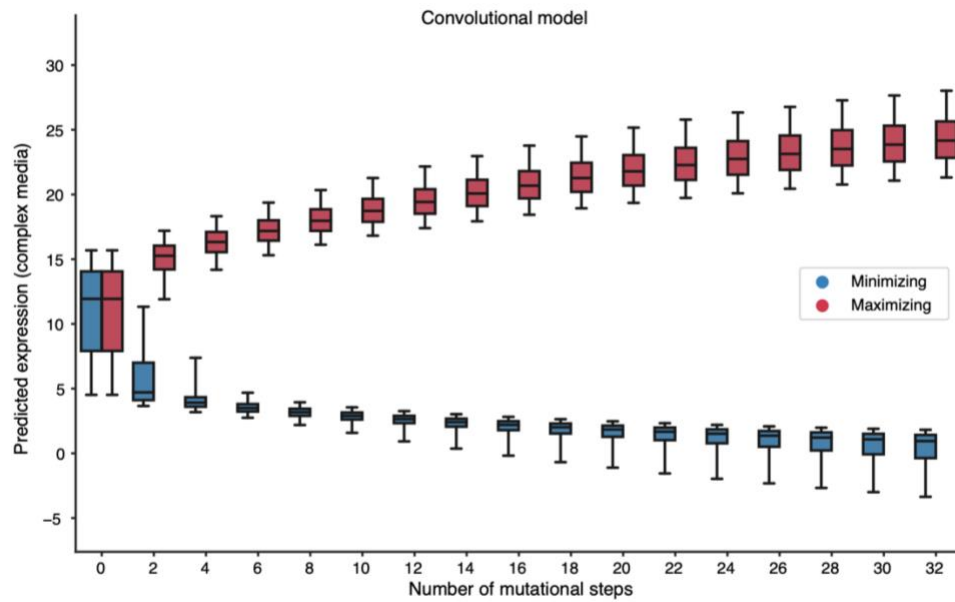


**Supplementary Data Fig. 17 | Characterization of sequence trajectories under strong competing selection pressures using the transformer model. a-d**, Competing expression objectives are slow to reach saturation. **a,b**, Difference in predicted expression (y axis) at each evolutionary time step (x axis) under selection to maximize (red) or minimize (blue) the difference between expression in defined and complex media, starting with either native sequences (**a**, n=5,720 trajectories) or random sequences (**b**, n=10,000 trajectories). **c-d**, Distribution of predicted expression (y axis) in complex (blue) and defined (red) media at each evolutionary time step (x axis) for a starting set of native sequences (**c**, n=5,720 trajectories) and random sequences (**d**, n=10,000 trajectories). Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range. **e** Motifs enriched within sequences evolved for competing objectives in different environments. Top five most enriched motifs, found using DREME(Bailey, 2011) (**Methods**) within sequences computationally evolved from a starting set of random sequences to either maximize (left) or minimize (right) the difference in expression between defined and complex media, along with DREME E-values, the corresponding rank of the same motif when using native sequences as a starting point, the likely cognate TF and that TF's known motif.

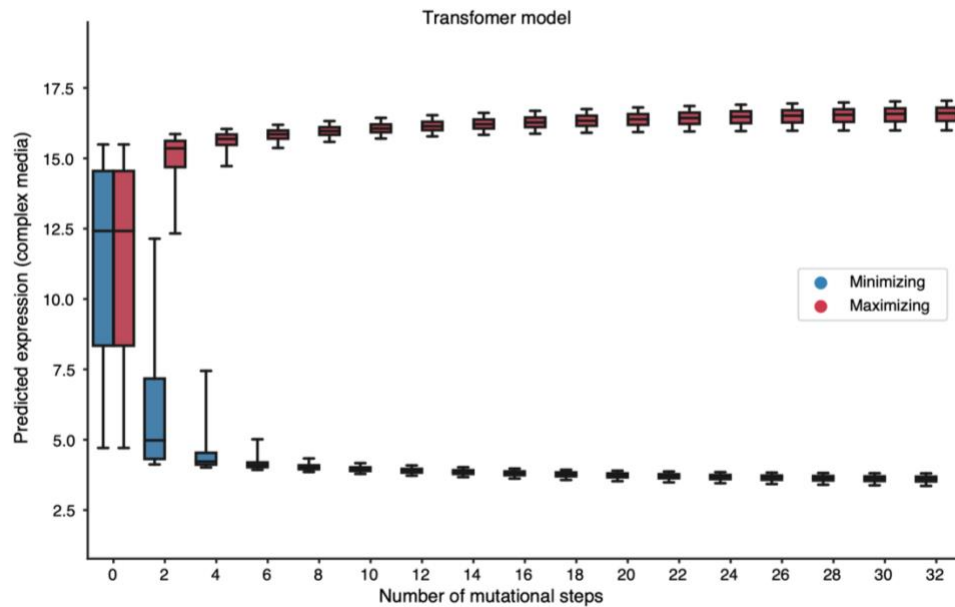


**Supplementary Fig. 18 | Sequences took diverse paths to evolve extreme expression in simulations with the transformer model. a-b,** The number (*y* axis) of mutations across each promoter position (*x* axis; -160 to -80 region) for trajectories under the SSWM regime starting with native promoter sequences when (a) maximizing or (b) minimizing expression in defined media using the transformer model. **c,d,** The number (*y* axis) of mutations of each type (*x* axis) for trajectories under the SSWM regime starting with native promoter sequences when (c) maximizing or (d) minimizing expression in defined media. Colors represent the mutational step (1-10).

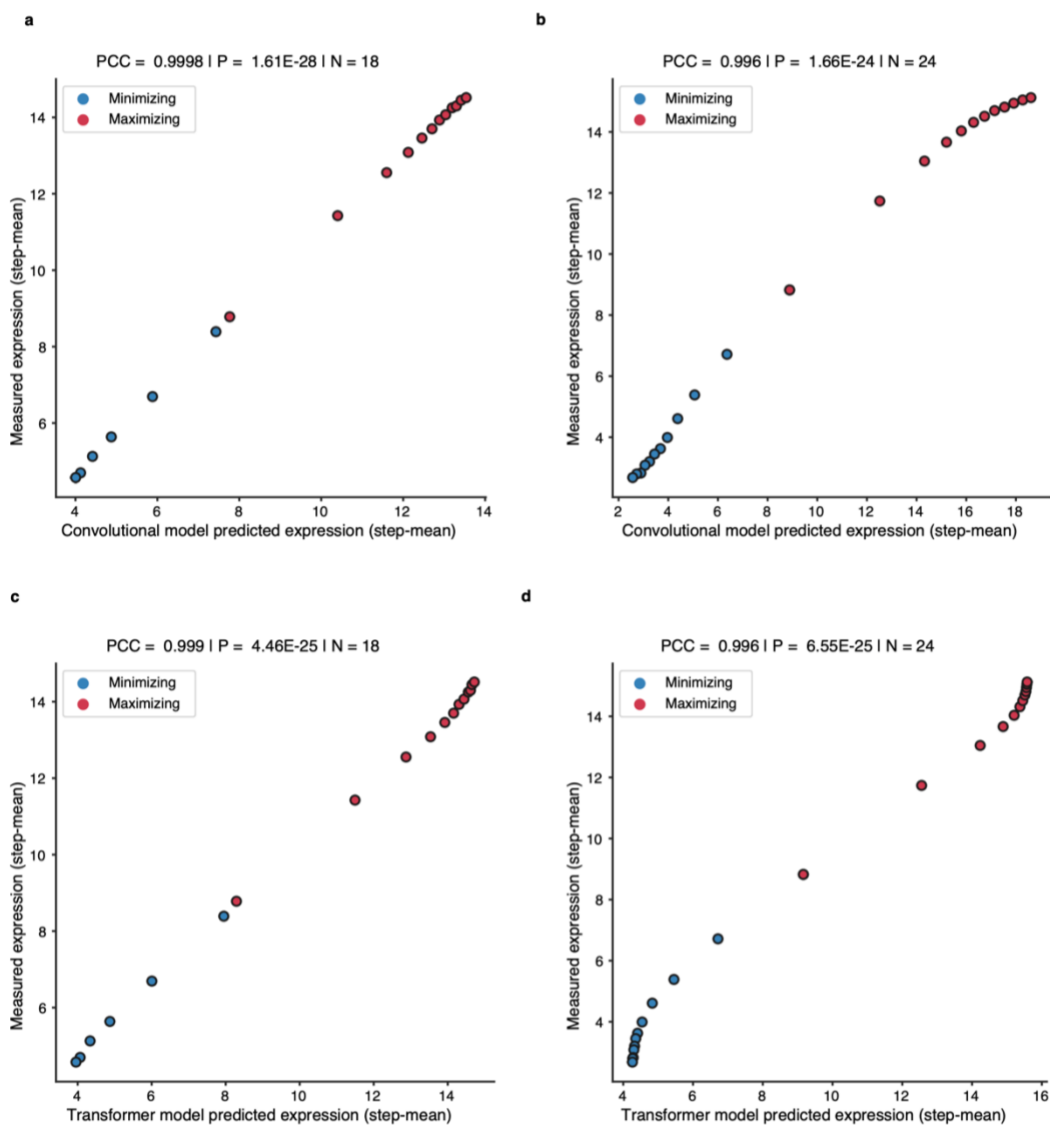
a



b

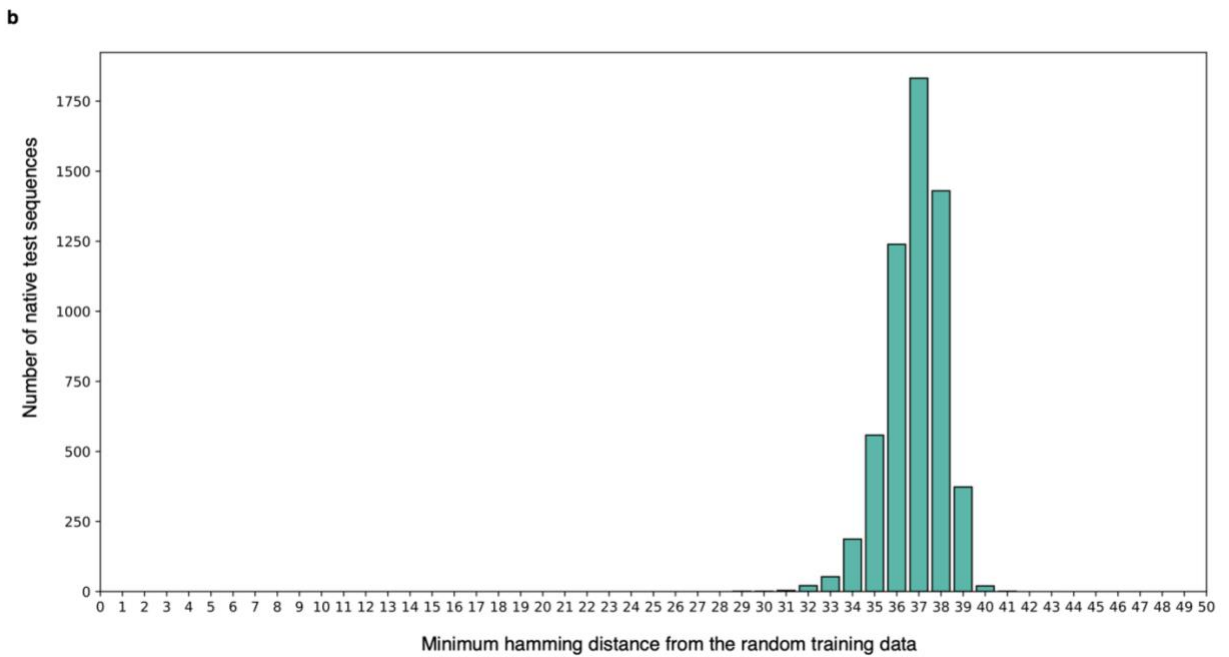
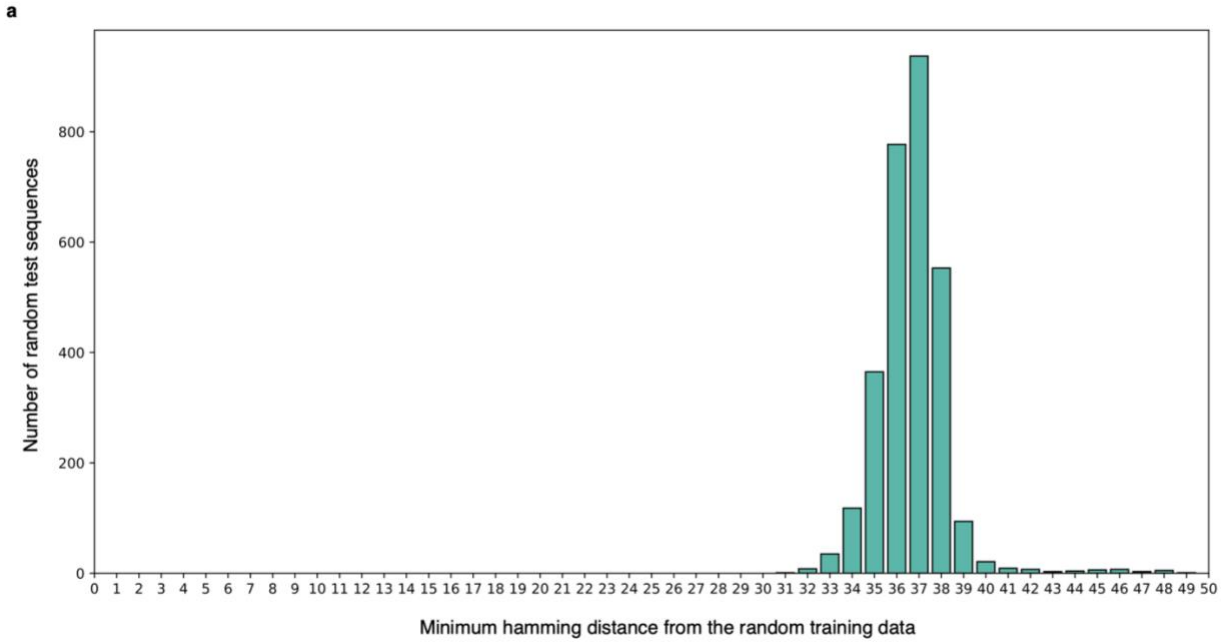


**Supplementary Fig. 19 | The transformer model captures expression plateau better than the convolutional model when simulating trajectories under SSWM for 32 mutational steps.** Distribution of predicted expression levels (y axis) in complex media at each mutational step (x axis) for sequence trajectories under SSWM favoring high (red) or low (blue) expression, starting with native promoter sequences using the convolutional (a, n=5,720 trajectories) or transformer (b, n=5,720 trajectories) models. The transformer model predicts an expression level plateau (like the measured expression in Fig. 2g), while the convolutional model predictions do not plateau at higher mutational distances. Midline: median; boxes: interquartile range; whiskers: 5<sup>th</sup> and 95<sup>th</sup> percentile range.



**Supplementary Fig. 20 | Summary statistic scatterplot for trajectories under SSWM.** Mean measured expression (y axis) and mean predicted expression (x axis) at each step in the mutational trajectories for native sequences under SSWM for the convolutional (**a,b**) and transformer (**c,d**) model in the complex (**b,d**) and defined (**a,c**) media, as in Supplementary Fig. 19. The Pearson's correlation coefficient (PCC) and the corresponding two-tailed p-value are shown.





**Supplementary Fig. 21 | Sequence differences between training and test data.** The distribution of the Hamming distance between each sequence in the (a) random or (b) native test sets and the closest sequence in in the random training set.

## Supplementary Tables

### Supplementary Tables

**Supplementary Table 1** | The Expression Conservation Coefficient (ECC), mutational robustness, evolvability vector archetypal coordinates, predicted expression, ECC using gene-specific correction factors, and ECC non-neutrality p-values corresponding to all native promoter sequences.

**Supplementary Table 2** | The GO terms enriched by the ECC ranking. One-sided p-values were computed using minimum hypergeometric statistics, taking into account multiple testing as previously described (Eden *et al.*, 2009).

Supplementary Tables 1 and 2 are provided as an Excel file.

**Supplementary Table 3 | Primers used in this study.** The list of single stranded oligonucleotides used. This table can be found in the Supplementary Information document.

Name	Sequence (5'-3')	Orientation	Description	Reference
pCDC36_DBVPG6765_WT_fw	ATCCATACACAAGACTCATAGAA	Fw	WE gRNA	This study
pCDC36_DBVPG6765_WT_rv	AACTTCTATGAGTCTTGTGTATG	Rv	WE gRNA	This study
D6765_to_Y12_ssODN	TTCCATCTCTATATAACAAAGTAT TTCTTTATTTTCTAATAGTTCCTTT CTACGAGTCTTGTGTATGTTTATA AAGAGTGAGCTCTTTTGTATGAA GT	Duplex	ssODN SA allele	This study
pCDC36_seq_F	TCACACGTAGACGACTTGCCA	Fw	Sequencing	This study
pCDC36_seq_R2	CCTTGTAGTTTTTGCATATCTAGT	Rv	Sequencing	This study
Seq_3_Fw	ACTTGCCACATCCTGGTGTT	Fw	Sequencing	This study
Seq_3_Rv	ATGTTTCTGCCCACGGTGAT	Rv	Sequencing	This study
CDC36_Fw	CATGACCTTAGGAGCGGACT	Fw	qPCR	This study
CDC36_Rv	TCCACTTCGCTTCTGGATGT	Rv	qPCR	This study
ACT1_Fw	TTGGCCGGTAGAGATTTGAC	Fw	qPCR	Teste et al.
ACT1_Rv	CCCAAAACAGAAGGATGGAA	Rv	qPCR	Teste et al.

RPN2_Fw	GCGGATACAGGCACATTGGATAC C	Fw	qPCR	Teste et al.
RPN2_Rv	TGTTGCTACCTTCTCTACCTCCTT ACC	Rv	qPCR	Teste et al.
pT-pA_GibsRI	GAACTGCATTTTTTTCACATCNNN NNNNNNNNNNNNNNNNNNNNNNNN NNNNNNNNNNNNNNNNNNNNNNNN NNNNNNNNNNNNNNNNNNNNNNNN NNNNNNNNNNNGGTTACGGCTGT TTCTTAA	Fw	Random promoter oligo for use in pTpA promoter context	de Boer et al
R-pT_GibsDS	TTAAGAAACAGCCGTAACC	Rv	For double- stranding pT- pA_GibsRI	de Boer et al
Nextera_i5LN5_GpT	TCGTCGGCAGCGTCAGATGTGTA TAAGAGACAGNNNNNTGCATTTT TTTCACATC	Fw	Nextera adaptor addition, with 5 random bases to help clustering	de Boer et al
Nextera_i7R_GpA	GTCTCGTGGGCTCGGAGATGTGT ATAAGAGACAGAACAGCCGTAAC C	Rv	Nextera adaptor addition	de Boer et al

**Supplementary Table 4 | Strains.** The list of yeast strains used.

Strain	Genotype	Reference
Y8205	<i>MATalpha, can1delta ::STE2pr-Sp_his5 lyp1delta ::STE3pr-LEU2 his3delta1 leu2delta0 ura3delta0</i>	Charles Boone Lab – strain verified by auxotrophy
S288C:: <i>ura3</i>	<i>MATα SUC2 gal2 mal2 mel flo1 flo8-1 hap1 ho bio1 bio6 ura3delta0</i>	de Boer et al. 2020 – strain verified by PCR of URA3
DBVPG6765 (WE)	<i>MATalpha, ho::NatMX, ura3::KanMX</i>	Cubillos, Louis & Liti (DOI: 10.1111/j.1567- 1364.2009.00583.x)
Y12 (SA)	<i>MATalpha, ho::NatMX, ura3::KanMX</i>	Cubillos, Louis & Liti
WE C7	DBVPG6765 derivate with SA Upc2 binding site	This study – pCDC36 genotype verified by Sanger sequencing
WE C23	DBVPG6765 derivate with SA Upc2 binding site	This study – pCDC36 genotype verified by Sanger sequencing

## References

- Agarwal V, Shendure J. Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks. 2020. *Cell Reports* 31 (7), 107663.
- Alipanahi, B. *et al.* (2015) “Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning,” *Nature Biotechnology*, 33(8), pp. 831–838.
- Bailey, T. L. (2011) “DREME: motif discovery in transcription factor ChIP-seq data,” *Bioinformatics (Oxford, England)*, 27(12), pp. 1653–1659.
- de Boer, C. G. *et al.* (2020) “Deciphering eukaryotic gene-regulatory logic with 100 million random promoters,” *Nature biotechnology*, 38(1), pp. 56–65.
- de Boer, C. G. and Hughes, T. R. (2012) “YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities,” *Nucleic Acids Res*, 40(Database issue), pp. D169–79.
- Brodsky, S. *et al.* (2020) “Intrinsically Disordered Regions Direct Transcription Factor In Vivo Binding Specificity,” *Molecular Cell*, 79(3), pp. 459–471.e4.
- Chen, J. *et al.* (2019) “A quantitative framework for characterizing the evolutionary history of mammalian gene expression,” *Genome research*, 29(1), pp. 53–63.
- De Boer, C. (2017) “High-efficiency *S. cerevisiae* lithium acetate transformation v1 (protocols.io.j4tcqwn),” *protocols.io*. ZappyLab, Inc. doi: 10.17504/protocols.io.j4tcqwn.
- Eden, E. *et al.* (2009) “GORilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists,” *BMC bioinformatics*, 10, p. 48.
- Keren, L. *et al.* (2016) “Massively Parallel Interrogation of the Effects of Gene Expression Levels on Fitness,” *Cell*, 166(5), pp. 1282–1294.e18.
- Langmead, B. *et al.* (2009) “Ultrafast and memory-efficient alignment of short DNA sequences to the human genome,” *Genome biology*, 10(3), p. R25.
- Li, J. *et al.* (2020) “DeepATT: a hybrid category attention neural network for identifying functional effects of DNA sequences,” *Briefings in bioinformatics*. doi: 10.1093/bib/bbaa159.
- Quang, D. and Xie, X. (2016) “DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences,” *Nucleic acids research*, 44(11), p. e107.
- Quang, D. and Xie, X. (2019) “FactorNet: A deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data,” *Methods (San Diego, Calif.)*, 166, pp. 40–47.
- Sharon, E. *et al.* (2012) “Inferring gene regulatory logic from high-throughput measurements of thousands of systematically designed promoters,” *Nature biotechnology*, 30(6), pp. 521–530.
- Shrikumar, A., Greenside, P. and Kundaje, A. (2017) “Reverse-complement parameter sharing improves deep learning models for genomics,” *bioRxiv*, p. 103663.

Vaswani, A. *et al.* (2017) “Attention is All you Need,” in Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 5998–6008.

Weirauch, M. T. *et al.* (2013) “Evaluation of methods for modeling transcription factor sequence specificity,” *Nature Biotechnology*, 31(2), pp. 126–134.

Weirauch, M. T. and Hughes, T. R. (2010) “Conserved expression without conserved regulatory sequence: the more things change, the more they stay the same,” *Trends in genetics: TIG*, 26(2), pp. 66–74.

Zhou, J. and Troyanskaya, O. G. (2015) “Predicting effects of noncoding variants with deep learning-based sequence model,” *Nature methods*, 12(10), pp. 931–934.

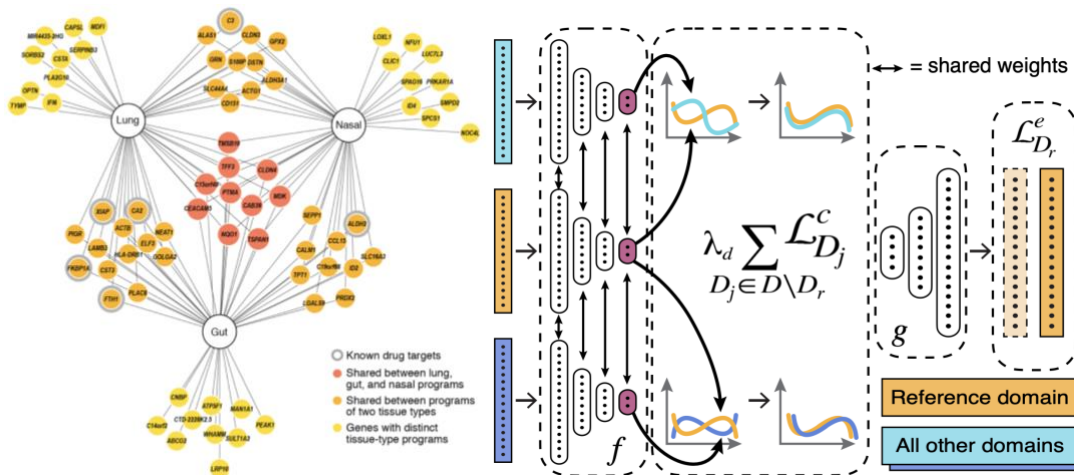




**Part B:**

*Expression*

## ATLAS (A Tool for Learning from Atlas-scale Single-cell measurements)



This chapter describes ‘A Tool for Learning from Atlas-scale Single-cell measurements’(ATLAS).

An early version of ATLAS appeared in:

*Single-cell meta-analysis of SARS-CoV-2 entry genes across tissues and demographics. **Nature Medicine** 27, 546–559 ([Muus, C. et al. 2021](#)). **Contribution:** Co-first author.*

Spatial mapping of cell types using ATLAS, was first presented in:

*Reference-based cell type matching of spatial transcriptomics data. **bioRxiv** 2022. ([Zhang et al](#)).*

**Contribution:** Co-author.

The manuscript that follows in this chapter, describes how ATLAS can be used to predict and prioritize drug targets from single cell atlases. **Contribution:** First author.

---

# ATLAS | How to predict and prioritize drug targets by expression program inference from single cell atlases

---

**Eeshit Dhaval Vaishnav**  
Massachusetts Institute of Technology  
edv@mit.edu

**Charles Comiter**  
Massachusetts Institute of Technology

**Ayshwarya Subramanian**  
Broad Institute of MIT and Harvard

**Karthik Jagadeesh**  
Broad Institute of MIT and Harvard

**Aviv Regev**  
Massachusetts Institute of Technology  
aviv.regev.sc@gmail.com

## Summary

The ongoing COVID-19 pandemic caused by the SARS-CoV-2 virus has sparked an urgent need for better understanding of its pathogenesis and identification of new drug targets to stem both viral infection and tissue-, organ- and body-level responses. Viral infection and host response begin at the single-cell level: the virus infects only specific cell types, while additional cell types respond to the intracellular signals triggered by infection. Integrative analyses of single-cell atlases can help identify both the specific cells involved and their therapeutically targetable programs. However, the complex and non-linear variability between measurements made across domains such as individuals, conditions, technological platforms and laboratories makes inference from these atlases challenging. Here, we introduce a novel framework for inference from atlas-scale scRNA-seq datasets. First, we learn biologically informative, domain-invariant feature representations for scRNAseq expression data by aligning domain distributions in a latent space through moment matching using a regularized autoencoder, correcting for undesirable variability across measurement domains. Then, we use these domain-invariant representations to identify gene programs with non-linear and combinatorial effects on phenotype using feature importance measures. We apply our framework to single cell atlases from autopsies and bronchoalveolar lavage samples from COVID-19 patients along with atlases from healthy individuals in the Human Cell Atlas to infer expression programs in SARS-CoV-2 target cells. We then predict and prioritize putative drug targets validated from independently published, drug repurposing and protein-protein interaction studies. This framework extends to both discrete (e.g. diseased vs healthy) and continuous (e.g. copy number variant scores) labels for scRNA-seq datasets and has broad applicability across a range of human diseases.

## 1 Contributions

We introduce a novel framework for predicting and prioritizing putative human disease drug targets from single-cell atlases that we call ATLAS (A Tool for Learning from Atlas-scale Single-cell datasets), with two primary contributions :

1. ATLASa : We propose a method for solving the scRNA-seq data-integration problem by using a higher order moment-matching regularizer with an AE to learn domain-invariant feature representations for scRNA-seq data.
  - We validate our approach on synthetic and real datasets and benchmark our performance against existing methods to demonstrate the quality of our learned representations.
  - We use these representations to identify and annotate cell types in a newly generated single-cell atlas from COVID-19 lung autopsies and a recently published lung atlas from healthy individuals [1].
2. ATLASb : We describe a simple approach for inferring gene expression programs from single-cell atlases using feature importance methods on models trained on our domain-invariant features that accounts for non-linear and combinatorial interactions between genes and their effects on phenotype.
  - We apply this approach to a recently published dataset of bronchoalveolar lavage fluid samples from COVID-19 patients [2] to infer cell-type specific gene expression programs
  - We identify drug targets, including ones that we were able to validate using independent experimental studies of SARS-CoV-2 protein-protein interaction (PPI) [3] and drug-repurposing [4]. We also demonstrate how to use our approach to prioritize putative drug target lists.

## 2 Approach

Our proposed method for learning domain-invariant features, ATLASa, is outlined in Figure 1a. We formulate this problem as a multi-target domain adaptation problem. The model architecture is that of an autoencoder comprised of an encoder  $f$  and a decoder  $g$  that are parametrized as a neural network with parameters  $\theta$ . The autoencoder has  $m$  streams with shared parameters as shown and each steam corresponds to an experimental domain.

**Input** : The model expects gene expression vectors for cells as the input. This input is supplied to the arm corresponding to the domain the cell is sampled from.

**Output** : The model outputs two domain invariant representations for each input  $x$ . The first output is  $f_\theta(x)$ , the latent feature representation learned by the encoder and, the second output is  $g_\theta(f_\theta(x))$ , a domain-invariant reconstruction of the input by the AE. We refer to this reconstruction in (ii) as the 'corrected' gene expression vector from here on out.

We minimize the pairwise domain discrepancy between an arbitrarily chosen reference domain and the rest by adapting the Central Moment Discrepancy (CMD) [5, 6] regularizer for matching the higher order central moments using order-wise moment differences for the latent feature space learned by  $f$ . The CMD regularizer is appropriate for this atlas-scale scRNA-seq data-integration problem because of its scalability (CMD computation is linear in the number of samples), minimal parameter sensitivity[5] and its ability to match non-Normal distributions.

### 2.1 Problem Formulation

Let  $D = \{D_j\}_{j=1}^m$  denote the set of scRNA-seq measurement domains.  $D_j = \{x_i^j\}_{i=1}^{n_j}$ , where each  $x_i^j$  refers to a single-cell gene expression vector measured in domain  $D_j$ . We first arbitrarily choose a reference domain  $D_r \in D$ . Then, for training, we sample mini-batches of size  $n_b$  from each domain  $X_j = \{x_i^j \in D_j \mid |X_j| = n_b\}$ . The objective function  $\mathcal{L}$  is a linear combination of the reference domain reconstruction loss and the pairwise CMD domain discrepancy losses for each non-reference domain w.r.t the reference domain.

$$\mathcal{L} = \mathcal{L}_{D_r}^e + \lambda_d \sum_{D_j \in D \setminus D_r} \mathcal{L}_{D_j}^c \quad (1)$$

$$\mathcal{L}_{D_r}^e(\theta) = \frac{1}{n_b} \sum_{x \in X_r} \|x - g_\theta(f_\theta(x))\|^2 \quad (2)$$

$$\mathcal{L}_{D_j}^c(\theta) = \frac{1}{|b-a|} \|E(f_\theta(X_j)) - E(f_\theta(X_r))\|_2 + \sum_{k=2}^K \frac{1}{|b-a|^k} \|C_k(f_\theta(X_j)) - C_k(f_\theta(X_r))\|_2 \quad (3)$$

where  $\|\cdot\|$  denotes the Euclidean norm,  $E(X) = \frac{1}{|X|} \sum_{x \in X} x$  is the empirical expectation vector, the parameter  $K$  is the bound on the order of the central moment terms,  $C_k(X) = E((x - E(X))^k)$  is the vector of all  $k^{th}$  order sample central moments,  $\mathcal{L}_{D_r}^c$  denotes the reference domain reconstructions loss and  $\mathcal{L}_{D_j}^c(\theta)$  denotes the domain discrepancy loss term for every other domain.

The objective function is minimized w.r.t.  $\theta$  using gradient based optimization methods.

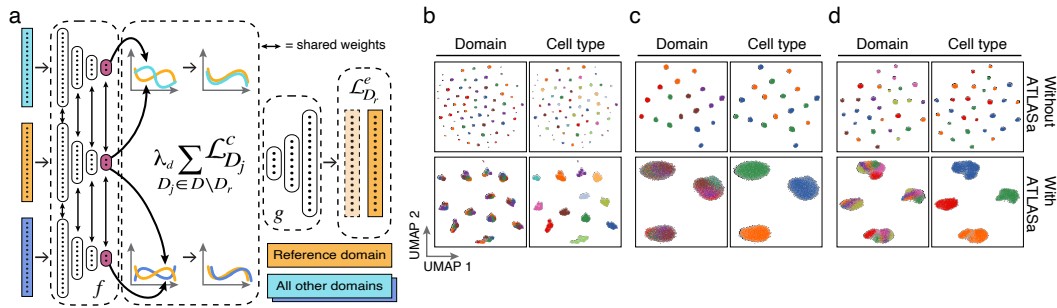
## 2.2 Datasets and Implementation Details

We use a letter code to refer to each dataset used in this manuscript. Complete details about accession (in the case of experimentally measured datasets) and simulations (in the case of synthetic datasets) will be made available with the Supplementary Materials. We used the following single-cell atlas datasets :

- COVIDA : A single-cell atlas of lung cells isolated from autopsy samples [citation TBD].
- COVIDB : A single-cell atlas of bronchoalveolar immune cells in COVID-19 patients [2].
- LUNG : A single-cell atlas of lung cells isolated from healthy individuals [1].
- SPLATTER2, SPLATTER4 and SPLATTER6 : Synthetic single-cell atlases that we generated for evaluation using a widely used scRNA-seq simulation library [7].
- SCIBSIM : A synthetic single-cell atlas used for benchmarking integration methods in [8].
- NASAL : A single-cell atlas generated from human surgical chronic rhinosinusitis tissue [9].

In addition to these single-cell atlases, we also used also used a dataset of SARS-CoV-2 protein-protein interactions (PPI) [3] and a drug-repurposing (REP) [4] study along with DRUGBANK [10], a comprehensive database of FDA approved and experimental drugs.

Details about the implementation, hyper-parameter considerations and dependencies can be found in the Supplementary Methods.



**Figure 1: ATLASa learns domain-invariant, biologically meaningful representations of single-cell data.** (a) A schematic outline of the ATLASa scRNA-seq data-integration method. Results of ATLASa applied to three simulated datasets (b, c, d) shown as 2-D UMAP visualizations. Each dot represents a cell, colored by domain of origin (left) or cell type (right). The top panels show the original simulated data UMAP visualizations without the ATLASa integration. The bottom panels show the same data with ATLASa integration. ATLASa integrated data shows clear separation of biologically relevant cell types when evaluated against the ground truth cell types used for the simulation, while displaying domain-invariance in the face of the original domain discrepancy.

### 3 Results

#### 3.1 Performance Evaluation and Benchmarking

First, we qualitatively evaluate the performance of our scRNA-seq data-integration method ATLASa using three synthetic single-cell atlases (SPLATTER2, SPLATTER4, SPLATTER6). Each of these atlases were simulated to have pre-determined domain and cell type labels for each cell in the simulation to serve as ground truth in order to assess whether ATLASa learned domain-invariant representations  $f_{\theta}(x)$  and whether these still retained biological information. We ran ATLASa on each of these simulated atlases to learn domain-invariant representations for each cell and visualized them using a 2-dimensional UMAP [11] projection with and without using their learned representations using ATLASa. Figures 1b, c, d clearly show that the representations learned by ATLASa preserve biological information, as demonstrated by their ability to separate out biologically relevant cell-types in an unsupervised manner.

**Table 1:** Summary of performance on (domain-invariance, biological information preservation) metrics of an array of integration methods, including our ATLASa method. Entries take the form (KBET, ILASW). Both metrics are set up such that they range from 0 to 1 and higher values reflect better performance.

<i>Integration</i> <i>Dataset</i>	ATLASa	scVI	Scanorama	Harmony	None
NASAL	(.632, .691)	(.343, .616)	(.390, .559)	(.568, .673)	(.556, .676)
SCIBSIM	(.999, .597)	(.916, .573)	(.997, .546)	(.735, .537)	(.743, .551)
COVIDB	(.283, .627)	(.219, .557)	(.379, .494)	(.200, .553)	(.350, .568)
SPLATTER6	(.969, .524)	(.233, .555)	(.140, .518)	(.936, .527)	(.362, .514)
AVERAGE	(.721, .610)	(.428, .575)	(.476, .529)	(.466, .569)	(.646, .580)

Next, we quantitatively establish the efficacy of ATLASa, our proposed data-integration method, through a comprehensive benchmarking analysis against an array of established data-integration methods: single-cell Variational Inference (scVI) [?], Scanorama [12], and Harmony [13] (we refer the reader to the Supplement for further description of these methods and a comprehensive report of the benchmarking analysis). Since there may exist a trade-off between learning domain-invariant representations and preserving biologically meaningful information, our benchmarking analysis used two complementary metrics: (i) the k-Nearest-Neighbors Batch Effect Test (KBET) [14], a specially designed scRNA-seq data-integration assessment metric for evaluating the representations learned by the methods on their domain-invariance, and (ii) the Isolated Label Average Silhouette Width (ILASW) score [8], a metric designed for evaluating the preservation of biologically relevant information. We set up both metrics such that they lie in the range in from 0 to 1 and such that higher values on each are indicative of better performance and we refer the reader to the Supplementary Material for a complete description of these details.

**Table 2:** Summary of runtimes (in seconds) for ATLASa’s data integration method vs. scVI.

<i>Integration</i> <i>Dataset</i>	ATLASa	scVI
NASAL	62.443	1023.624
SCIBSIM	70.817	1105.963
COVIDB	428.046	2917.076
SPLATTER6	110.230	1808.429
AVERAGE	167.884	1713.773

We ran our analysis on four datasets: two synthetic (SCIBSIM, SPLATTER6) and two biological (NASAL, COVIDB) (see **Datasets** and Supplementary Methods for further information). We found that on 3 out of 4 datasets, ATLASa’s data integration method outperformed the other methods on the domain-invariance metric (see **Table 1**). Furthermore, on 3 out of 4 datasets including the COVIDB dataset, ATLASa’s data integration method outperformed the others on the biological-information preservation metric. Importantly, ATLASa also had the highest average metric score on both of these complementary tasks (see **Table 1**).

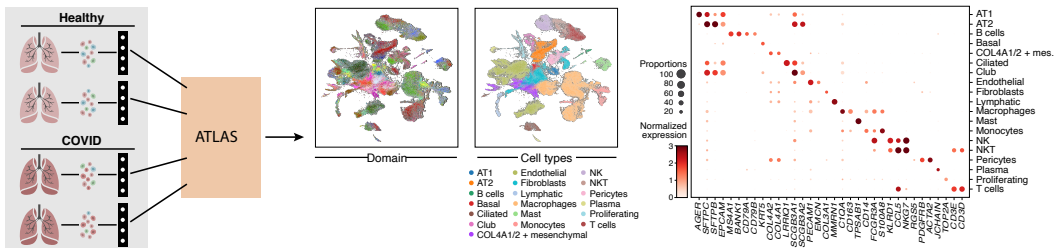
Next, we benchmark the runtime of our method. At the time of benchmarking, out of all the methods considered, only scVI and ATLASa have GPU implementations. We compared their runtimes on each of the four datasets (see **Table 2**). Compared to scVI, ATLASa is over ten times faster on

average (in addition to being more effective on the domain-invariance and biological-information preservation tasks on all and all but one datasets, respectively, as shown in **Table 1**). One of the reasons for the runtime performance is the fact that CMD does not require computationally expensive kernel computations.

However, all of these metrics computed for various benchmarking tasks are just proxies for demonstrating potential utility of these approaches on real-life problems. Now, we turn our attention to a very real-life problem : COVID-19.

### 3.2 Identification and Annotation of Cell Types in a new COVID-19 Patient Single-Cell Atlas using Domain-Invariant representations from ATLASa

We use ATLASa to integrate two single cell atlases collected at opposite ends of the phenotype continuum: (i) A healthy human lung single cell atlas [11] spanning 17 experimental domains and 60,872 cells and (ii) A new lung cell atlas created from autopsies of COVID-19 patients ([citation pending]) representing 15 experimental domains and 27,519 cells. We then use the  $f_\theta(x)$  learned by ATLASa to identify and annotate cell types from these tissues to better understand COVID-19 biology using unsupervised clustering [Supplementary Materials]. We annotated the atlases using our results post-hoc using literature derived markers (Figure 2d) to define a shared taxonomy of 19 cell classes as shown in Figure 2c. The shared taxonomy we report includes Epithelial (AT1, AT2, Basal, Ciliated and Club cells), Stromal (Endothelial, Lymphatic, Mesenchymal) and Immune (Macrophages, Monocytes, Mast, T and B) cells. Further, Figure 2b demonstrates the domain-invariance of the learned representations from a diverse set of individuals displaying a broad range of phenotypes.

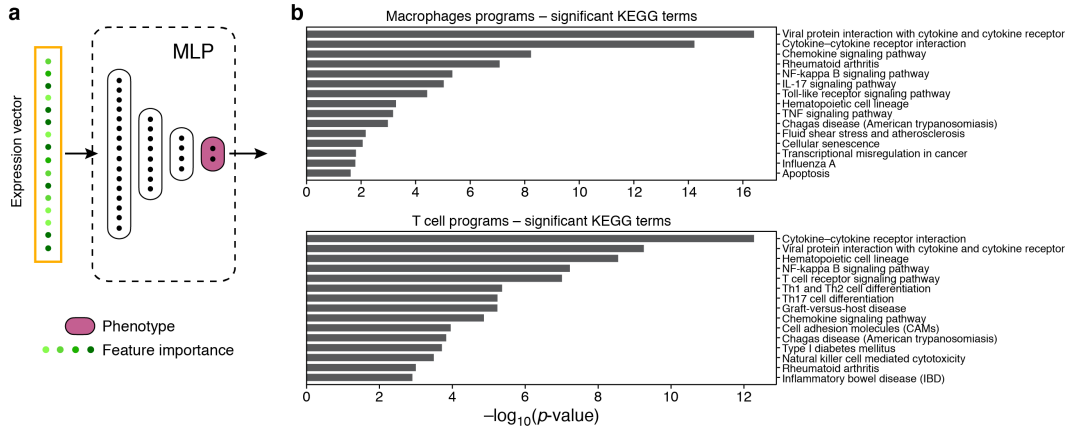


**Figure 2:** ATLASa identifies cell types in a new COVID-19 single-cell atlas. Demonstration of ATLASa for jointly analyzing two atlases. (a) Lung samples from (i) healthy and (ii) COVID-19 autopsy cohorts serve as input to ATLASa. Each lung sample represents an experimental domain. 2-D UMAP representations of the ATLAS integrated cells (dots) colored by domain of origin (b) and putative cell type (c). (d) Dotplot representation of marker genes used to annotate cell types. The size of the dot represents the proportion of cells expressing the marker gene (columns) and the color represents the average normalized gene expression in each cell type (rows). Normalized gene expression values are capped at 3.

### 3.3 Gene Expression Program Inference for COVID-19 from Healthy and Diseased Tissue Atlases

Genes can affect phenotypes in non-linear and combinatorial ways. Understanding these gene-expression programs (GEPs) that drive phenotypes of interest  $P$  can help elucidate the molecular mechanisms and pathways corresponding to disease states and facilitate drug target identification. Previous attempts at inferring GEPs for COVID19 relied heavily on distinguishing  $ACE2+TMPRSS2+$  cells (double positives, DPs) and compared these cells to  $ACE2-TMPRSS2-$  cells (double negatives, DNs) [15]. However these approaches were limited in their ability to identify GEPs relevant for drug target identification because of a lack of availability of single-cell atlases from COVID-19 patients. We use COVIDB, a recently published atlas of BALF samples from COVID-19 [2], to describe and demonstrate a simple approach (that we call ATLASb) for GEP inference from single-cell atlases.

ATLASb takes the ‘corrected’ gene expression vectors  $g_\theta(f_\theta(x))$  for each cell  $x$  produced by ATLASa as input to train a simple multi-layer perceptron (MLP) model  $F$  to predict phenotype  $P$  from  $g_\theta(f_\theta(x))$  such that  $F(g_\theta(f_\theta(x))) = P$ . Since the dimensions of the input of  $F$  correspond to a domain-invariant representations of gene expression vectors, we can now employ importance measures like SHAP values [16] and DeepLIFT [17] to identify gene expression programs driving



**Figure 3:** (a) Identifying GEPs using feature importance measures with MLPs. Gene programs inferred from the COVIDB dataset. KEGG gene set enrichment of *severe* (b) Macrophage and (c) T-cell expression programs. X-axis represents the enrichment test log p-values after adjustment for multiple hypothesis testing. Y-axis represents the enriched gene sets.

the phenotype of interest. This simple framework can allow us to identify genes that have non-linear and combinatorial effects on phenotypes because of the use of an MLP as the model  $F$ .

For the COVIDB dataset, we used a gradient-based approximation of SHAP values [16] with ATLASb to identify cell type specific GEPs over three different labeling regimes as phenotypes : *severity*, whether a cell is from a healthy patient or one with a severe case of COVID-19, *DP*, whether a cell has non-zero ACE2 and TMPRSS2 expression levels or zero for both, and *viral*, whether a cell has been infected by the virus or is simply a non-infected bystander [Supplementary Methods].

The GEPs allowed us to identify shared and cell type specific gene expression features ( Figure 3c, d). Overall, expression programs in all immune cells were characterized by enrichment for classic inflammatory molecules including *IL6*, members of the TNF family and complement component 3 (*C3*). The *severity* expression programs in macrophages were strongly enriched in cytokines and chemokines indicative of a pro-inflammatory “cytokine storm-like” state consistent with what has previously been reported [2]. On the other hand, the *viral* expression programs of macrophages included interferon regulatory genes *IRF8*, *IRF4* consistent with response to viral entry. Cell-type specific *severity* programs in epithelial cells include genes known to mediate viral infection (*CEACAM5*, *CLDN4*, *CLDN1*), and multi-functional cytokines including *IL10* and *CD40* signaling previously reported in inflamed airway cells [18]. Interestingly, *TMPRSS2* emerged as a highly discriminant gene, supporting its suggested role [19] as an accessory protease in mediating viral entry.

Taken together, this strong validation of our gene expression programs corroborated by multiple independent publications demonstrates the efficacy of our proposed ATLAS framework and its potential for driving biological discovery.

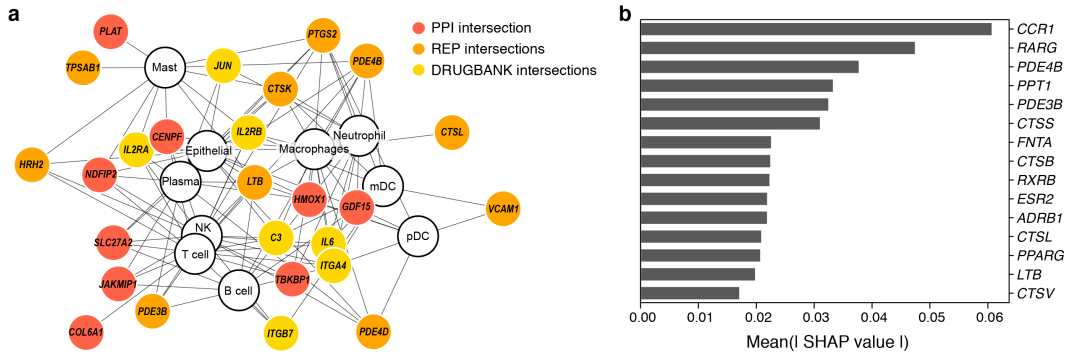
### 3.4 Predicting and Prioritizing Putative Drug Targets for COVID-19

Finally, since we had all the pieces in place to do so, we sought to evaluate whether our hypothesis that “domain-invariant feature representations of scRNA-seq data can help address the pressing need for identification of human disease drug targets when used in tandem with modern feature importance methods to account for the combinatorial and non-linear effects of gene-expression on phenotype” holds true in the context of COVID-19. We identified two independent validation datasets from a growing body of work on drug-screens and protein-protein interaction studies for COVID-19 : a dataset of SARS-CoV-2 protein-protein interactions (PPI) [3] and another from a SARS-CoV-2 drug-repurposing (REP) [4] study. We found multiple overlapping genes between our gene expression programs and the lists of drug targets identified in both of these independent experimental studies shown in Figure 4a. We also show in Figure 4a that our results hold across cell types suggesting that the putative drugs may potentially have mechanisms of action that could make them work across cell types. Figure 4a also shows a subset of its overlapping genes with Drugbank [10], a comprehensive



database of FDA approved and experimental drugs which may suggest novel putative targets with known drug-target interaction that were identified from scRNA-seq data using ATLAS and may have been missed by PPI studies potentially because of their indirect mechanism of action. Further inspection of Figure 4a shows that besides having the known contender IL6 which serves as further validation of our approach, the target lists also contain other putative hits including the complement component C3, and the inflammation-induced mediator of tissue tolerance GDF15 [20]. Other interesting drug targets include CDK6, which has not been previously reported in the context of SARS-CoV-2 but it has been reported as an interactor for viral cyclins [21].

These results suggest that our hypothesis may hold true.



**Figure 4:** (a) Network plot showing a subset of the drug targets discovered by ATLAS and their source of independent experimental validation : DRUGBANK [10] (a comprehensive list of FDA approved and experimental drug targets) intersections shown in yellow, COVID-19 REP (a large scale SARS-CoV-2 drug repurposing study) intersections in orange and the COVID-19 PPI (protein-protein interactions) intersections in red. (b) Prioritizing putative drug targets identified in the COVID-19 REP publication by ranking them using their feature SHAP importance values for predicting disease phenotypes.

For the development of effective therapeutics for COVID-19, putative drug targets must be prioritized in concert with the cellular context and function. Cell-type specific gene expression programs from COVID-19 data afford one window into both cellular localization and putative functional context for such prioritization. We show how one can prioritize the drug targets identified by the REP study (Figure 4b) by layering on the information from the COVIDB single-cell atlase using the same simple ATLASb framework described above. Among the top hits from this prioritization approach were Cathepsins *CTSS*, *CTSB*, *CTSL*, lysosomal genes essential for the cellular entry of coronaviruses [22], and previously proposed as targets for SARS-CoV [23]. Recently, amantadine was identified as an inhibitor of *CTSL*, and proposed [24] as a potential therapeutic for COVID-19 as well. Taken together, these independent sources from literature suggest that our approach for prioritizing putative COVID-19 drug targets may provide useful information for further follow-up with experimental studies.

## References

- [1] Kyle J Travaglini, Ahmad N Nabhan, Lolita Penland, Rahul Sinha, Astrid Gillich, Rene V Sit, Stephen Chang, Stephanie D Conley, Yasuo Mori, Jun Seit, Gerald J Berry, Joseph B Shrager, Ross J Metzger, Christin S Kuo, Norma Neff, Irving L Weissman, Stephen R Quake, and Mark A Krasnow. A molecular cell atlas of the human lung from single cell RNA sequencing. August 2019.
- [2] Mingfeng Liao, Yang Liu, Jing Yuan, Yanling Wen, Gang Xu, Juanjuan Zhao, Lin Cheng, Jinxiu Li, Xin Wang, Fuxiang Wang, Lei Liu, Ido Amit, Shuye Zhang, and Zheng Zhang. Single-cell landscape of bronchoalveolar immune cells in patients with covid-19. *Nature Medicine*, page 174, May 2020.
- [3] David E Gordon, Gwendolyn M Jang, Mehdi Bouhaddou, Jiewei Xu, Kirsten Obernier, Kris M White, Matthew J O’Meara, Veronica V Rezelj, Jeffrey Z Guo, Danielle L Swaney, Tia A Tummino, Ruth Huettnerhain, Robyn M Kaake, Alicia L Richards, Beril Tutuncuoglu, Helene

- Foussard, Jyoti Batra, Kelsey Haas, Maya Modak, Minkyu Kim, Paige Haas, Benjamin J Polacco, Hannes Braberg, Jacqueline M Fabius, Manon Eckhardt, Margaret Soucheray, Melanie J Bennett, Merve Cakir, Michael J McGregor, Qiongyu Li, Bjoern Meyer, Ferdinand Roesch, Thomas Vallet, Alice Mac Kain, Lisa Miorin, Elena Moreno, Zun Zar Chi Naing, Yuan Zhou, Shiming Peng, Ying Shi, Ziyang Zhang, Wenqi Shen, Ilsa T Kirby, James E Melnyk, John S Chorba, Kevin Lou, Shizhong A Dai, Inigo Barrio-Hernandez, Danish Memon, Claudia Hernandez-Armenta, Jiankun Lyu, Christopher J P Mathy, Tina Perica, Kala B Pilla, Sai J Ganesan, Daniel J Saltzberg, Ramachandran Rakesh, Xi Liu, Sara B Rosenthal, Lorenzo Calviello, Srivats Venkataramanan, Jose Liboy-Lugo, Yizhu Lin, Xi-Ping Huang, Yongfeng Liu, Stephanie A Wankowicz, Markus Bohn, Maliheh Safari, Fatima S Ugur, Cassandra Koh, Nastaran Sadat Savar, Quang Dinh Tran, Djoskhun Shengjuler, Sabrina J Fletcher, Michael C O’Neal, Yiming Cai, Jason C J Chang, David J Broadhurst, Saker Klippsten, Phillip P Sharp, Nicole A Wenzell, Duygu Kuzuoglu, Hao-Yuan Wang, Raphael Trenker, Janet M Young, Devin A Cavero, Joseph Hiatt, Theodore L Roth, Ujjwal Rathore, Advait Subramanian, Julia Noack, Mathieu Hubert, Robert M Stroud, Alan D Frankel, Oren S Rosenberg, Kliment A Verba, David A Agard, Melanie Ott, Michael Emerman, Natalia Jura, Mark von Zastrow, Eric Verdin, Alan Ashworth, Olivier Schwartz, Christophe d’Enfert, Shaeri Mukherjee, Matt Jacobson, Harmit S Malik, Danica G Fujimori, Trey Ideker, Charles S Craik, Stephen N Floor, James S Fraser, John D Gross, Andrej Sali, Bryan L Roth, Davide Ruggero, Jack Taunton, Tanja Kortemme, Pedro Beltrao, Marco Vignuzzi, Adolfo Garcia-Sastre, Kevan M Shokat, Brian K Shoichet, and Nevan J Krogan. A SARS-CoV-2 protein interaction map reveals targets for drug repurposing. *Nature*, April 2020.
- [4] Laura Riva, Shuofeng Yuan, Xin Yin, Laura Martin-Sancho, Naoko Matsunaga, Sebastian Burgstaller-Muehlbacher, Lars Pache, Paul P De Jesus, Mitchell V Hull, Max Chang, Jasper Fuk-Woo Chan, Jianli Cao, Vincent Kwok-Man Poon, Kristina Herbert, Tu-Trinh Nguyen, Yuan Pu, Courtney Nguyen, Andrey Rubanov, Luis Martinez-Sobrido, Wen-Chun Liu, Lisa Miorin, Kris M White, Jeffrey R Johnson, Christopher Benner, Ren Sun, Peter G Schultz, Andrew Su, Adolfo Garcia-Sastre, Arnab K Chatterjee, Kwok-Yung Yuen, and Sumit K Chanda. A large-scale drug repositioning survey for SARS-CoV-2 antivirals. April 2020.
- [5] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for Domain-Invariant representation learning. February 2017.
- [6] Werner Zellinger, Bernhard A Moser, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Robust unsupervised domain adaptation for neural networks via moment alignment. *Inf. Sci.*, 483:174–191, May 2019.
- [7] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: Simulation of single-cell rna sequencing data. *Genome Biology*, page 174, September 2017.
- [8] M D Luecken, M Büttner, K Chaichoompu, A Danese, M Interlandi, M F Mueller, D C Strobl, L Zappia, M Dugas, M Colomé-Tatché, and F J Theis. Benchmarking atlas-level data integration in single-cell genomics. May 2020.
- [9] Jose Ordovas-Montanes, Daniel F Dwyer, Sarah K Nyquist, Kathleen M Buchheit, Marko Vukovic, Chaarushena Deb, Marc H Wadsworth, 2nd, Travis K Hughes, Samuel W Kazer, Eri Yoshimoto, Katherine N Cahill, Neil Bhattacharyya, Howard R Katz, Bonnie Berger, Tanya M Laidlaw, Joshua A Boyce, Nora A Barrett, and Alex K Shalek. Allergic inflammatory memory in human respiratory epithelial progenitor cells. *Nature*, 560(7720):649–654, August 2018.
- [10] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.*, 46(D1):D1074–D1082, January 2018.
- [11] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. February 2018.
- [12] Bonnie Berger Brian Hie, Bryan Bryson. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Naure Biotechnology*, 37(June):685–691, 2019.

- [13] Ilya Korsunsky, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-Ru Loh, and Soumya Raychaudhuri. Fast, sensitive, and accurate integration of single cell data with harmony.
- [14] Maren Büttner, Zhichao Miao, F Alexander Wolf, Sarah A Teichmann, and Fabian J Theis. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods*, 16(1):43–49, January 2019.
- [15] Christoph Muus, Malte D Luecken, Gokcen Eraslan, Avinash Waghay, Graham Heimberg, Lisa Sikkema, Yoshihiko Kobayashi, Eeshit Dhaval Vaishnav, Ayshwarya Subramanian, Christopher Smilie, Karthik Jagadeesh, Elizabeth Thu Duong, Evgenij Fiskin, Elena Torlai Triglia, Meshal Ansari, Peiwen Cai, Brian Lin, Justin Buchanan, Sijia Chen, Jian Shu, Adam L Haber, Hattie Chung, Daniel T Montoro, Taylor Adams, Hananeh Aliee, J Samuel, Allon Zaneta Andrusivova, Ilias Angelidis, Orr Ashenberg, Kevin Bassler, Christophe Bécavin, Inbal Benhar, Joseph Bergensträhle, Ludvig Bergensträhle, Liam Bolt, Emelie Braun, Linh T Bui, Mark Chaffin, Evgeny Chichelnitskiy, Joshua Chiou, Thomas M Conlon, Michael S Cuoco, Marie Deprez, David S Fischer, Astrid Gillich, Joshua Gould, Minzhe Guo, Austin J Gutierrez, Arun C Habermann, Tyler Harvey, Peng He, Xiaomeng Hou, Lijuan Hu, Alok Jaiswal, Peiyong Jiang, Theodoros Kappellos, Christin S Kuo, Ludvig Larsson, Michael A Leney-Greene, Kyungtae Lim, Monika Litviňuková, Ji Lu, Leif S Ludwig, Wendy Luo, Henrike Maatz, Elo Madisson, Lira Mamanova, Kasidet Manakongtreecheep, Charles-Hugo Marquette, Ian Mbano, Alexi Marie McAdams, Ross J Metzger, Ahmad N Nabhan, Sarah K Nyquist, Lolita Penland, Olivier B Poirion, Sergio Poli, Cancan Qi, Rachel Queen, Daniel Reichart, Ivan Rosas, Jonas Schupp, Rahul Sinha, Rene V Sit, Kamil Slowikowski, Michal Slyper, Neal Smith, Alex Sountoulidis, Maximilian Strunz, Dawei Sun, Carlos Talavera-López, Peng Tan, Jessica Tantivit, Kyle J Travaglini, Nathan R Tucker, Katherine Vernon, Marc H Wadsworth, Julia Waldman, Xiuting Wang, Wenjun Yan, William Zhao, Carly G K Ziegler, The NHLBI LungMAP Consortium, and The Human Cell Atlas Lung Biological Network. Integrated analyses of single-cell atlases reveal age, gender, and smoking status associations with cell type-specific expression of mediators of SARS-CoV-2 viral entry and highlights inflammatory programs in putative target cells. April 2020.
- [16] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.
- [17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. April 2017.
- [18] Francesca Cagnoni, Susanna Oddera, Julien Giron-Michel, Anna Maria Riccio, Susanna Olsson, Palmiro Dellacasa, Giovanni Melioli, G. Walter Canonica, and Bruno Azzarone. Cd40 on adult human airway epithelial cells: Expression and proinflammatory effects. *The Journal of Immunology*, 172(5):3205–3214, 2004.
- [19] Markus Hoffmann, Hannah Kleine-Weber, Simon Schroeder, Nadine Krüger, Tanja Herrler, Sandra Erichsen, Tobias S. Schiergens, Georg Herrler, Nai-Huei Wu, Andreas Nitsche, Marcel A. Müller, Christian Drosten, and Stefan Pöhlmann. Sars-cov-2 cell entry depends on ace2 and tmprss2 and is blocked by a clinically proven protease inhibitor. *Cell*, 181(2):271 – 280.e8, 2020.
- [20] Harding H. Luan, Andrew Wang, Brandon K. Hilliard, Fernando Carvalho, Connor E. Rosen, Amy M. Ahasic, Erica L. Herzog, Insoo Kang, Margaret A. Pisani, Shuang Yu, Cuiling Zhang, Aaron M. Ring, Lawrence H. Young, and Ruslan Medzhitov. Gdf15 is an inflammation-induced central mediator of tissue tolerance. *Cell*, 178(5):1231 – 1244.e11, 2019.
- [21] Ursula Schulze-Gahmen and Sung-Hou Kim. Structural basis for CDK6 activation by a virus-encoded cyclin. *Nat. Struct. Biol.*, 9(3):177–181, March 2002.
- [22] Christine Burkard, Monique H Verheije, Oliver Wicht, Sander I van Kasteren, Frank J van Kuppeveld, Bart L Haagmans, Lucas Pelkmans, Peter J M Rottier, Berend Jan Bosch, and Cornelis A M de Haan. Coronavirus cell entry occurs through the endo-/lysosomal pathway in a proteolysis-dependent manner. *PLoS Pathog.*, 10(11):e1004502, November 2014.
- [23] Graham Simmons, Dhaval N. Gosalia, Andrew J. Rennekamp, Jacqueline D. Reeves, Scott L. Diamond, and Paul Bates. Inhibitors of cathepsin l prevent severe acute respiratory syndrome

coronavirus entry. *Proceedings of the National Academy of Sciences*, 102(33):11876–11881, 2005.

- [24] Sandra P Smieszek, Bart P Przychodzen, and Mihael H Polymeropoulos. “amantadine disrupts lysosomal gene expression; potential therapy for COVID19”. April 2020.

---

# Supplementary Information

## ATLAS | How to predict and prioritize drug targets by expression program inference from single cell atlases

---

### Contents

<b>1 Introduction to the Supplementary Materials</b>	<b>1</b>
<b>2 Definitions</b>	<b>2</b>
<b>3 Datasets</b>	<b>2</b>
<b>4 Implementation, Preprocessing and Hyperparameters</b>	<b>3</b>
<b>5 Performance Evaluation and Benchmarking</b>	<b>6</b>
<b>6 Gene Expression Program Inference for COVID-19 from Healthy and Diseased Tissue Atlases</b>	<b>7</b>
<b>7 Errata and Clarifications</b>	<b>10</b>

## 1 Introduction to the Supplementary Materials

In this document, we elaborate on the details omitted from the main text of the manuscript. Additionally, we will provide further clarification on the methods and results. Source code for ATLAS and all the analyses reported in this manuscript can be found in the files attached in the zipped folder (with the specific references to the files below). The datasets used are made available on a [Google Bucket](#) (with the exception of COVIDA, which is currently not publicly available). We begin by briefly summarizing our primary results :

In the main text, we introduced ATLAS: A Tool for Learning from Atlas-scale Single-cell datasets. ATLAS is a two part framework consisting of ATLASa and ATLASb. We used ATLASa to learn domain-invariant representations of scRNA-seq datasets and demonstrated the method's effectiveness qualitatively (in the main text's Figure 1) and quantitatively (in the main text's Table 1 and Table 2) using real and simulated datasets. Then, we demonstrated ATLASa's applicability to COVID-19 by using it to identify cell types from single-cell atlases constructed from autopsies of COVID-19 patients by integrating them with atlases from healthy control lung samples from [\[1\]](#). Next, we developed ATLASb, an approach for inferring gene-expression programs driving a phenotype (like disease state) using scRNA-seq atlases. Using ATLASb in tandem with ATLASa on a recently published COVID-19 single-cell atlas, we identified gene expression programs and drug targets, both of which we validated using multiple independent published studies.

## 2 Definitions

In this section, we define the terms we used throughout the main text and the supplement :

- **COVID-19** : Coronavirus Disease 2019
- **SARS-CoV-2** : Severe Acute Respiratory Syndrome Coronavirus 2
- **Gene** : A sequence of DNA characters called nucleotides that codes for the synthesis of gene products like proteins.
- **Protein** : A gene product that is a polymer made from amino acids that often has a defined structure and function.
- **Phenotype** : A set of observable traits of an organism that is a function of an organism’s gene expression profile and its environment.
- **Single-Cell Sequencing** : A set of technologies developed for making measurements from individual cells.
- **Single-cell RNA Sequencing (scRNA-seq)** : A technology that quantifies the gene expression from individual cells by measuring the amount of messenger RNA produced by all the genes in the cell.
- **Drug Targets** : A molecule in the body (often a protein, which is the product of the expression of a gene) that is strongly associated with a disease phenotype and that may be perturbed by a drug to produce a therapeutic effect.
- **Gene Expression Program (GEP)** : In this context, the set of genes that drive the corresponding phenotype of interest.
- **KEGG term** : KEGG (Kyoto Encyclopedia of Genes and Genomes) [2] is a widely used database for interpreting genomic data. KEGG terms refer to the molecular functions of individual genes and sets of genes (e.g. : a GEP).
- **FDA** : The Food and Drug Administration, a United States federal executive department responsible for the process of approving drugs.

## 3 Datasets

Here, we elaborate on the dataset descriptions provided in the main text. All simulations and published experimental datasets we used are made available as `.h5ad AnnData` objects [3] that can be found at [https://console.cloud.google.com/storage/browser/atlas\\_datasets/](https://console.cloud.google.com/storage/browser/atlas_datasets/). The name of each available dataset in the list below may be clicked-on for a direct download link to a pre-processed version of each dataset with the non-preprocessed, raw counts in the `AnnData.layers['counts']` field :

- [SPLATTER6](#), [SPLATTER4](#), [SPLATTER2](#) | Synthetic single-cell atlases that we generated for evaluation using a widely used scRNA-seq simulation tool `SpLatteR` [4]. `SPLATTER6` (shown in Figure 1b in the main text), `SPLATTER4` (shown in Figure 1c in the main text), and `SPLATTER2` (shown in Figure 1d in the main text) consisted of 23148, 19318, and 15441 cells, respectively, each with 9987, 9983, and 9974 genes, respectively. The datasets were simulated to have come from six, six and eight experimental domains, respectively and to have twelve, three, and four celltypes, respectively. The `AnnData.obs.Batch` field contains the domain label for each cell (in each of these three datasets) and similarly the `AnnData.obs.Group` field contains the cell type label for each cell.
- [NASAL](#) | A single-cell atlas generated from human surgical chronic rhinosinusitis tissue [5]. It consists of 7087 cells, each with 33694 measured genes. The `AnnData.obs.donor` field contains the domain label for each cell and the `AnnData.obs.ann_level_4` field contains the cell type label for each cell.
- [SCIBSIM](#) | A synthetic single-cell atlas used for benchmarking integration methods based on simulations from [6] (their manuscript contains further details on how they simulated this dataset).

It contains 20100 cells, each with 10000 genes. The `AnnData.obs.Batch` field contains the domain label for each cell and the `obs.Group` field contains the cell type labels for each cell.

- **COVIDB** | A recently published single-cell atlas of bronchoalveolar immune cells in COVID-19 patients [7]. It consists of 63103 cells, each with 33538 genes. The `AnnData.obs.sample_new` field contains the domain label for each cell and the `AnnData.obs.celltype` field denotes the cell type (out of the ten celltypes reported in the publication) that each cell belongs to.
- Datasets used in Figure 3 : We used two single-cell atlases for Figure 3. The first (COVIDA) is from an ongoing effort to generate atlases from lung autopsies of COVID-19 patients and the second (LUNG) is an atlas generated from lung samples isolated from healthy individuals.
  - COVIDA - A single-cell atlas of lung cells isolated from autopsy samples [citation TBD]. It consists of 34523 cells, each with 28560 genes. We will make a data download link available for this dataset when the full dataset and the associated citation becomes publicly available. We will also update this section of the supplement with a final citation (currently pending) when it is possible to appropriately attribute everyone who generated this growing single-cell atlas.
  - LUNG - A single-cell atlas of lung cells isolated from healthy individuals [1]. It consists of 60993 cells, each with 26485 genes. This dataset is available at <https://hlca.ds.czbiohub.org/>.

## 4 Implementation, Preprocessing and Hyperparameters

### Implementation

All code used is made available along with the Supplementary Materials in the ATLAS-master folder. The `README.md` file in this folder describes how to create an environment for using ATLAS with all the dependencies installed. A description of each file follows :

- `tables_12main.ipynb` : Notebook to produce corrected `AnnData` objects and performance metrics for ATLASa and a suite of other data-integration methods (from Table 1 and Table 2 of the main manuscript).
- `program_analysis.ipynb` : Notebook to produce cell-type specific gene programs for a variety of feature-importance and phenotype labeling regimes used in the analysis including Tables 1, 2 and 3 shown in this Supplement.
- `tables_123456supp_figures_4supp.ipynb` : Notebook to produce Tables 4, 5 and 6 in this Supplement and to find the intersections for the Venn Diagram in this Supplement's Figure 4.
- `figures_12main_123supp.ipynb` : Notebook to reproduce hyperparameter experiments, pre vs post ATLASa-correction UMAPs, and dot-plot associated with Figures 1 and 2 in the main text and Figures 1, 2, and 3 in this Supplement.
- `figures_34main.ipynb` : Notebook to produce the GEP bar-plots and drug-target network plot from Figures 3 and 4 in the main text of the manuscript.
- `correct_aux.py` : Backend of the ATLASa data-integration method and associated helper functions.
- `analysis_aux.py` : Backend of the ATLASb feature-importance drug-target-prediction method and associated helper functions.
- `atlas.py` : Programmer friendly API for ATLAS use on other scrRNA-seq atlases and disease labels (currently under development, to be officially released soon).

The ATLASa and ATLASb model implementations used `Tensorflow` [8] and `Keras` [9]. The training and evaluation were carried out on a NVIDIA Tesla M60 GPU. A machine with consistent hardware configurations was used for all the benchmarking experiments. All single-cell RNA-sequencing datasets were converted to `.h5ad AnnData` objects [3] before preprocessing as described below. The primary language of all the implementation was Python, with the exception of the simulated scrRNA-seq datasets that used R to interface with Splatter [4]. The network plot in Figure 4 was generated using

`networkx` and the KEGG term enrichment was performed using the `scanpy.queries.enrich` function in Scanpy [10].

## Preprocessing

For use as input with ATLASa, we started with raw count matrices from each atlas dataset above whenever those were available. Then, we normalized the counts such that the sum of expression values across genes for each cell was  $10^4$ . Then we log scaled the data after adding 1 to each entry in the gene expression matrix. When raw counts weren't available, we directly used the dataset with the transcripts per million (TPM) units made available with the published dataset. After this step, for all the datasets, we then normalized again to a target sum of 1, zero-centered and scaled the data to have mean 0 and unit variance, and clipped off the scaled values to restrict them to a maximum value of 10. We used the same preprocessing steps for all the external data-integration methods we benchmarked our method against with the exception of the methods that required raw, un-processed counts to be presented as input (scVI, in particular). The domain from which the largest number of cells originated was arbitrarily chosen as the Reference Domain  $D_r$  for training the model. When working with large-scale datasets, ATLASa can use the projection of the gene expression data along its first hundred orthonormal principal component vectors before the first layer of the AE and then use the corresponding inverse transform at the end of the output layer of the AE for speed and scalability. This option was used whenever the `hp_dict['use_rep']` value was set to 'X\_pca' in the code.

For use as input to ATLASb, we directly used the corrected gene expression vectors output by ATLASa for each cell as described in the main text. We note that the simple approach suggested in ATLASb is independent of preprocessing. Any form of the gene expression data may be used to train a classifier from gene expression matrices to predict phenotypes and this classifier may then be examined using variable importance measures to identify GEPs and drug targets as described in the main text. Here, the dimensions of the input correspond to the genes measured in the cell. To speed up ATLASb, one can use a subset of the genes that are highly variable (defined using the `scanpy.pp.highly_variable_genes` function) and/or the subset on the genes that are upregulated in expression (defined as the genes that have a higher mean expression in one phenotype class than another) in addition to differentially expressed genes (identified using the `scanpy.tl.rank_genes_groups` function). Finally, for the COVIDB dataset, the publication reported some ambient contamination (more details can be found in their [Data exclusions](#) section of their original publication [7]) and so we addressed that by adapting a procedure previously used in [11] for filtering out putative ambient RNA contaminants. This procedure gave us a list of contaminant genes for each cell type that we subsequently excluded from all further analysis using ATLASb on the COVIDB dataset.

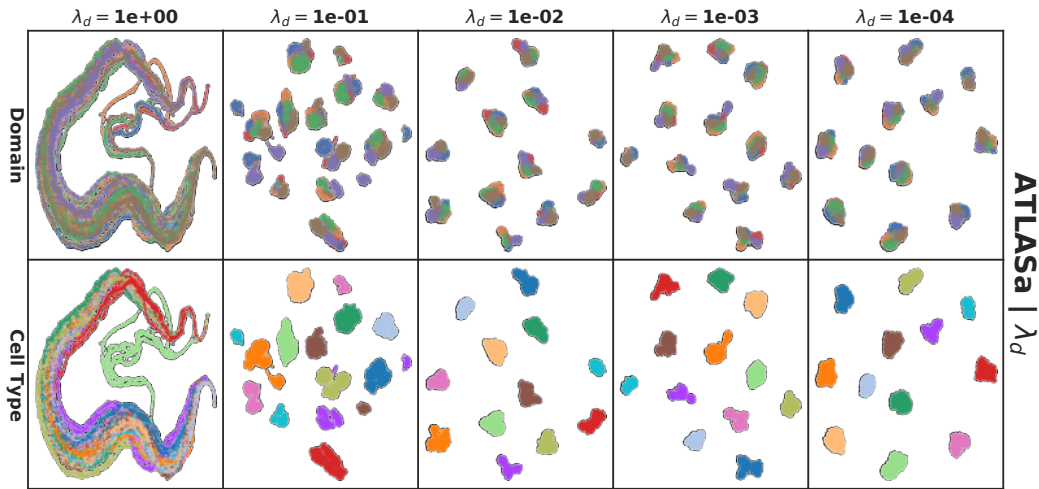
## Hyperparameters

The model architecture consists of an autoencoder (AE) with multiple streams, each corresponding to a domain as described in Section 3 of the main text. The encoder consists of three fully-connected layers with [1024, 512, 258] neurons respectively. The decoder consists of two fully connected layers, the first of which has 512 neurons and the second one has the same dimensions as the input. Each fully connected layer (except the last one) is followed by a Parametric ReLu Activation layer [12] and each connection has a 0.1 probability of dropout. The reconstruction loss term on the cells from the reference domain is computed using the `mean_squared_error` function. For every other domain, the domain discrepancy term w.r.t. the reference domain is computed using the `cmd` function adapted from [13]. The objective function is constructed as described in the main text and is optimized using the Adam Optimizer [14]. The model was trained for 10 epochs with a mini-batch size of 32. The code corresponding to this section can be found in the `get_corrected_adata` function from the `correct_aux.py` file shared with the Supplementary Materials. The `hp_dict` variable for each dataset described in the code specifies the exact hyperparameters used for each experiment.

The hyperparameter  $\lambda_d$ , the weight of the sum of the distribution moment matching terms for each domain, was chosen after a series of experiments on the SPLATTER datasets. We started with a low value for  $\lambda_d$  and increased the value by an order of magnitude until it broke the model at  $\lambda_d = 1$  (evident from the lack of separation between the ground truth cell types in the simulation for  $\lambda_d = 1$  in Figures 1, 2 and 3 in this supplement). The range considered was [1e-4, 1e-3, 1e-2, 1e-1, 1

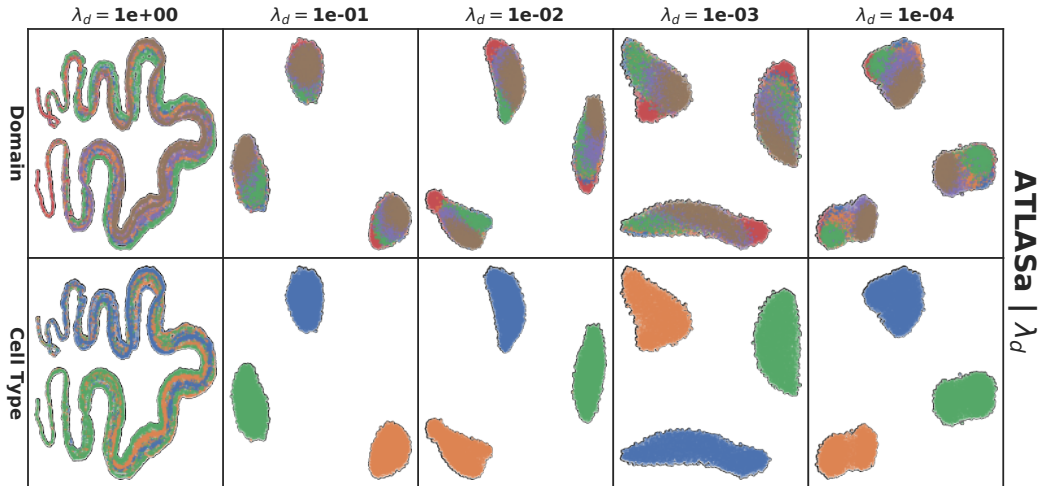


1



**Figure 1:** A range of  $\lambda_d$  values for SPLATTER6 with ATLASa

1



**Figure 2:** A range of  $\lambda_d$  values for SPLATTER4 with ATLASa

and all values of  $\lambda_d < 1$  performed very well on the domain-invariant representation learning task as shown in this Supplement's Figure 1, Figure 2 and Figure 3 by the separation of cell types across domains. The first row in each of these figures shows the cells colored by their experimental domain of origin and the second row shows cells colored by their biological cell type pre-defined in the simulation. Each column corresponds to a unique value of  $\lambda_d$ . We picked the middle of this range  $\lambda_d = 1e-2$  for all the subsequently analyzed COVID-19 relevant datasets whose results are presented in the manuscript. The uncorrected version of the data can be found in the top row of the main text Figures 1b, c and d. The bottom row of the main text Figures 1b, c and d were generated from scratch after the hyperparameter exploration using  $\lambda_d = 1e-2$ .

Finally, the hyperparameter  $K$ , the bound on the order of the moment central moment terms, is chosen as  $K = 3$  based on the hyperparameter sensitivity experiments in [13] showing that their results weren't sensitive to the choice of  $K$  for  $K \geq 3$ . The value of  $K = 3$  remained unchanged for all the results shown throughout.

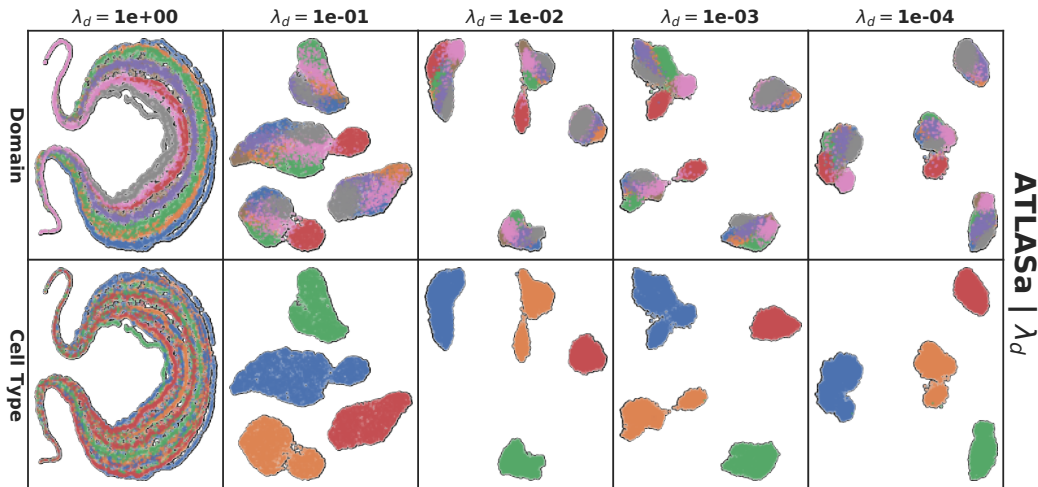


Figure 3: A range of  $\lambda_d$  values for SPLATTER2 with ATLASa

## 5 Performance Evaluation and Benchmarking

In this section, we elaborate on the benchmarking procedure for quantitatively establishing the efficacy of our ATLASa method for learning domain-invariant representations. First, we provide a summary of the external published data-integration methods that we compared our method against (all hyper-parameters for external published data-integration methods were influenced by a recent benchmarking paper [15]) :

- single-cell Variational Inference (scVI) [16]: a correction method that employs a variational autoencoder (VAE) [17] with a Bayesian hierarchical model to embed a gene expression vector, represented by a negative binomial distribution conditioned on batch-labels and measurement-specific variables, to a lower dimensional latent space. scVI was shown to be particularly effective on non-synthetic datasets with intricate effects from domain-misalignment [15]. scVI requires raw-counts, free of any pre-processing, as inputs, which we ensured to be accessible by storing such counts with every dataset. We used scVI Version 0.5.0. Regarding hyper-parameter choices, we trained a negative-binomial based VAE, with two hidden layers in the encoder and decoder, with dimensions 128 and 30, with learning rate .001, for a number of epochs  $\max(400, 8000000 \div \text{number-of-cells})$ , on a single NVIDIA Tesla M60 GPU. Their corrected embedding existed in a 30-dimensional latent space.
- Scanorama: Scanorama [18] is a panorama-sketching inspired integration technique that treats experimental data from different domains as different, smaller "snapshots" of a larger atlas-panorama. It takes these snapshots, aligns different cell types from all different pairs of samples, and "sketches" these snapshots together into a single embeded scRNA-seq "panorama" hyperplane. Scanorama was too shown to be, like scVI, particularly effective on non-synthetic datasets with intricate effects from domain-misalignment [15]. We utilized Version 1.4 of Scanorama with default hyper-parameters.
- Harmony: Harmony [19] is an iterative correction method that takes the principal components of each of the sampled atlases and produces a corrected embedding. Until convergence, Harmony first creates cell clusters of maximum batch-label diversity and then fits an appropriate linear mixture model according to different batches. Harmony was found perform better on synthetic than real biological atlases by [15]. We utilized Version 1.0 of Harmony with default hyper-parameters.

In comparing against these, ATLASa was trained with hyperparameters as defined by the `hp_dict` dictionary in the `tables_12main.ipynb`. The same values for the hyperparameters as specified in the dictionary were used for all the benchmarking analyses.

Now, we elaborate on the metrics we used in the benchmarking section to quantitatively measure the efficacy of ATLASa. All metrics have been adapted to take on values between 0 and 1, with higher values being indicative of better performance for easier interpretation. Furthermore, all metrics were considered on a 'corrected' embedding of the data output by each method ('corrected' refers to the domain-invariant representation of the data learned by each method in this context). We used two metrics, one that evaluated the learned representations on their domain-invariance and their effectiveness is correcting for domain-specific effects and the other metric evaluated the representations on the degree of conservation of biological information :

- K-Nearest-Neighbors Batch Effect Test: kBET [20], is a standard metric for measuring the biases introduced by the fact that the datasets are collected across a vast array of experimental domains. It is used to determine the similarity between the domain-label distribution of cells' nearest neighbors and that of the global atlas. In the original publication, kBET takes on a value between 0 to 1 such that a high kBET value is indicative of more drastic batching effects (in the original publication, higher values were worse). A robust data integration method would thus yield an atlas with a low kBET value. The method first divides the cells into sub-datasets according to cell-type. Then, a kBET value is calculated for each cell type. This is done by running using python's rpy2 library and rpy2.robjests to call the kBET function from R's kBET package. Subsequently, the mean kBET value is found across these cell types. To yield the metric we report as specified above, we subtract said mean from 1 to arrive at a final value for our reported metric and refer to it as kBET in the text. This metric is also used in [6] for evaluating performance on the same task.
- Isolated Label ASW: ILASW, developed by [15], determines how well an integration method accommodates celltypes/labels that are isolated: those not found in every sample. For each isolated label, we first assign, to each datapoint, a binary indicator label with respect to the isolated label. We find the Average Silhouette Width [21], a measure of cluster quality ranging from -1 (intersecting and poorly formed) to 1 (discrete and well formed) with respect to this indicator label using sklearn.metrics's silhouette\_score API and scale it with  $(1 + ASW) \div 2$  to take a value in-between 0 and 1. We lastly take the mean of this over all isolated label names. This describes how well sample-isolated information is retained post-integration. This metric is also used in [6] for evaluating performance on the same task.

## 6 Gene Expression Program Inference for COVID-19 from Healthy and Diseased Tissue Atlases

In constructing Gene Expression Programs for our COVID-19 atlases, we considered three different phenotype labelling regimes:

- Severity: whether a cell is from a healthy patient or one with a severe case of COVID-19.
- Double Positive (DP): whether a cell has non-zero ACE2 and TMPRSS2 expression levels or zero for both.
- Virality: whether a cell has been infected by the virus or is simply a non-infected bystander.

Moreover, we considered multiple different approaches for assigning feature importance measures to construct the gene-expression programs. In all of these approaches, we partitioned the cells according to their cell type. Then, we made a 75:25 train/test split and trained a classifier to predict a phenotype label within each cell type. The top five hundred most important genes were used as the ranked list of genes to define the GEP driving the corresponding phenotype. The classification approaches and corresponding variable importance measures we used were :

- Multi-Layer Perceptron (MLP) with SHAP Values: Training a simple multi-layer perceptron as a classifier. The MLP transformed an input gene expression vector to a 1000 dimensional feature vector using a fully-connected layer, followed by another layer with 100 nodes, followed by the output layer. The classifier was trained using stochastic gradient descent, with the number of epochs ranging between 10 and 16 and batch size between 32 and 256 depending on the training-atlas size (see analysis\_aux.py for further details). We then used the shap package from [22] to derive SHAP value inspired feature importances (see below for a brief explanation of SHAP values and their application in this context).

- We considered both `shap.DeepExplainer` (Deep Shap) and `shap.GradientExplainer` (Grad Shap) (see below again for further explanation).
- In initializing these explainers, we considered two background datasets: a random sample of 1000 training features (which, according to [22], provides a highly accurate estimate of true SHAP values) and the whole dataset. Similarly, for computing the SHAP values themselves, we considered two similar settings: computation using a random sample of 1000 test features or alternatively, using the entire test set. In each setting, we computed the appropriate SHAP values and, following [22], found feature importance by ranking features according to the largest average absolute value of the corresponding entry over the computed SHAP values.
- We found that using the GradientExplainer (Grad Shap), with the background dataset as the whole train-atlas, and computing SHAP values from the whole test-atlas, gave optimal results (as measured by correspondence with independent publications as described in the main text). We thus, used this feature-importance scheme going forwards for this classification approach. The results used in the main text used Grad Shap with these settings due to its efficiency and large observed (REP, PPI, DRUGBANK) intersection sizes on a per-cell-type basis. For Deep Shap, while we used the entire test-atlas to produce SHAP values, we were limited by their implementation to using a 1000-sample of train-atlas gene-expression vectors as a background dataset for the `shap.DeepExplainer` object.
- Random Forest: Using sklearn's [23] `RandomForestClassifier` to classify cells. We set all hyper-parameters to default settings. We computed the Mean Decrease in Impurity for the Random Forest classifier. We used these values from the `feature_importances_` attribute of the classifier in order to determine importances and used them to identify the GEPs.
- Gradient Boosted Decision Trees: Using sklearn's [23] `GradientBoostingClassifier` to classify cells. We set all hyper-parameters to default settings. We again used the `feature_importances_` attribute of the classifier in order to determine importances and used them to identify the GEPs.

### SHAP values and the use of shap with ATLASb

This section contains a brief overview of SHAP (SHapley Additive exPlanations) values [22] and how they are computed in the context of ATLAS. We first consider a machine learning model  $M : \mathbb{R}^n \rightarrow \mathbb{R}$  and a "background" dataset  $B$ ; additionally, let  $e_M = \mathbb{E}_{x \in B} M(x)$ . Given a gene expression vector  $g \in \mathbb{R}^n$ , the corresponding SHAP value of  $g$ ,  $S(g) \in \mathbb{R}^n$ , is one such that  $(\sum_{i=1}^n S(g)_i) + e_M = M(g)$ .  $S(g)$  lets us know how much each feature in  $g$  contributed to the value of  $M(g)$  being different from  $e_M$ . Thus, a SHAP value gives us a way to see how *important* each feature is.

To determine feature importances with the MLP classifier used in ATLASb, we leveraged the `shap` package of Lundberg and Lee [22]. `shap` features an array of model explainers, each with its own `Explainer.shap_values` method that takes in a list of  $m$  gene expression vectors and returns the corresponding SHAP values for each. Each of these explainers takes a model and background dataset as inputs and uses these to in their own computations of SHAP values. We considered two Explainers with our MLP classifier :

- `shap.DeepExplainer`, which approximates SHAP values for neural nets via an extension of DeepLIFT [24].
- `shap.GradientExplainer`, which uses a combination of Integrated Gradients [25], SHAP values, and SmoothGrad [26] as a measure of feature importance.

To evaluate how similar the results from the various classification and feature importance approaches are, we used the full list of all enriched terms returned by the `scanpy.queries.enrich` function when queried on the corresponding GEPs from each of the three methods. Then we compared their similarity using the Szymkiewicz-Simpson coefficient, also known as the overlap coefficient in Tables 1, 2 and 3 below for each feature-importance method used in each class of

phenotype labelling regime. The overlap coefficient for two sets  $X, Y$  is given by  $\frac{|X \cap Y|}{\min(|X|, |Y|)}$ . The Tables 1, 2 and 3 show the overlap coefficient of the GEPs found by other methods when compared to the Grad Shap importance method (for each phenotype labeling scheme and for each cell-type in the labeling scheme). Entries take on values between 0 and 1, with higher values indicating greater similarity between programs.

Tables 4, 5 and 6 describe the size of the intersection between the drug targets found by each of the feature-importance methods used and the (REP,PPI,DRUGBANK) datasets. The entries in the table are shown here as a tuple of (REP intersection size, PPI intersection size, DRUGBANK intersection size).

Finally, Figure 4 contains a Venn diagram showing that the results from all the feature-importance regimes used with ATLASb are remarkably similar lending further credence to the approach and indicating that the drug targets identified may deserve further follow up experiments.

**Table 1: Severity**

<i>Importance</i> \ <i>Cell Type</i>	B	NK	Mast	Macrophages	pDC	Epithelial	mDC	T	Plasma
Grad Shap	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Deep Shap	0.55	0.57	0.77	0.53	0.66	0.67	0.71	0.77	0.89
RF	0.74	0.78	0.54	0.64	0.64	0.61	0.67	0.78	0.70
GBT	0.63	0.85	0.63	0.68	0.54	0.63	0.66	0.71	0.74

**Table 2: Virality**

<i>Importance</i> \ <i>Cell Type</i>	T	Macrophages	Plasma	NK	Epithelial	Neutrophil	...	Epithelial
Grad Shap	1.00	1.00	1.00	1.00	1.00	1.00	...	1.00
Deep Shap	0.52	0.57	0.73	0.66	0.72	0.71	...	0.81
RF	0.53	0.63	0.58	0.69	0.66	0.83	...	0.58
GBT	0.66	0.40	0.39	0.62	0.58	0.65	...	0.65

**Table 3: DP**

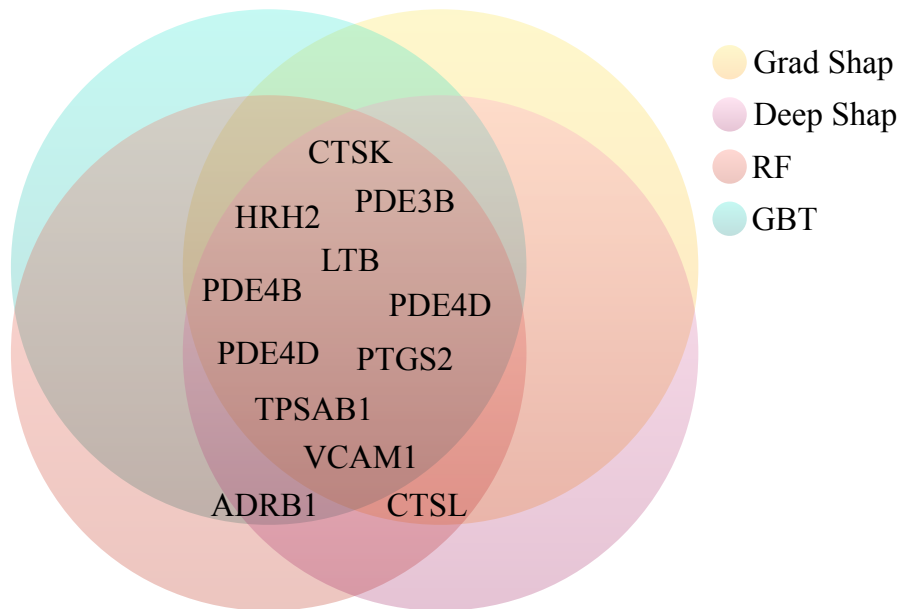
**Table 4: Severity**

<i>Importance</i> \ <i>Cell Type</i>	B	NK	Mast	Macrophages	pDC	Epithelial	mDC	T	Plasma
Grad Shap	(4, 1, 41)	(4, 0, 41)	(2, 5, 31)	(4, 2, 34)	(3, 4, 31)	(2, 4, 26)	(6, 3, 41)	(1, 4, 37)	(1, 2, 30)
Deep Shap	(2, 4, 38)	(4, 1, 42)	(5, 0, 35)	(7, 3, 43)	(2, 5, 34)	(4, 3, 38)	(1, 5, 30)	(4, 4, 31)	(2, 3, 27)
RF	(4, 3, 27)	(2, 4, 35)	(2, 3, 40)	(0, 2, 22)	(5, 2, 31)	(2, 1, 30)	(2, 3, 26)	(1, 5, 33)	(1, 2, 21)
GBT	(1, 2, 27)	(1, 2, 23)	(2, 4, 22)	(3, 0, 31)	(3, 3, 27)	(0, 1, 34)	(2, 4, 31)	(3, 1, 35)	(2, 2, 32)

**Table 5: Virality**

<i>Importance</i> \ <i>Cell Type</i>	T	Macrophages	Plasma	NK	Epithelial	Neutrophil	...	Epithelial
Grad Shap	(3, 3, 38)	(4, 4, 38)	(4, 3, 42)	(3, 0, 24)	(4, 2, 40)	(5, 0, 34)	...	(3, 2, 33)
Deep Shap	(4, 3, 38)	(3, 0, 24)	(4, 2, 37)	(6, 1, 31)	(5, 3, 35)	(4, 4, 38)	...	(3, 1, 31)
RF	(3, 3, 31)	(3, 5, 32)	(3, 4, 38)	(3, 1, 34)	(2, 3, 33)	(3, 0, 24)	...	(3, 1, 28)
GBT	(1, 1, 32)	(3, 0, 24)	(1, 3, 36)	(2, 4, 28)	(4, 1, 37)	(2, 0, 24)	...	(2, 1, 25)

**Table 6: DP**



**Figure 4:** The union, over all phenotype labeling schemes, of drug targets found for all four importance regimes that intersect with those found by the REP study (an independent experimental study on drug re-purposing for COVID-19). We note that all classification/feature-importance methods give an extremely similar results, implying that ATLASb isn't particularly sensitive to the feature importance method used and that the results are robust to that choice.

## 7 Errata and Clarifications

In this section we list typos and errors we spotted in our paper submission between the Paper Submission Deadline and the Supplementary Material Submission Deadline. We apologize for these errors and will make the necessary corrections (along with other errors we identify later in addition to suggested changes by reviewers) in the camera ready version of the paper.

- The legends for Figure 2b in the main text (the first UMAP plot from the left) should say 'Domain' instead of 'Batch'. This error was because 'Batch' and 'Domain' are used interchangeably in the scRNA-seq field, however we use 'Domain' throughout the text so as to not confuse it with 'mini-batch' used to train the model using stochastic gradient descent.
- In the main text, Figures 2a, 2b, 2c, 2d and 3c were not labelled with an appropriately "abcd" panel letter reference. We apologize for this error.
- We apologize for not defining KEGG terms in the main text.

## References

- [1] Kyle J Travaglini, Ahmad N Nabhan, Lolita Penland, Rahul Sinha, Astrid Gillich, Rene V Sit, Stephen Chang, Stephanie D Conley, Yasuo Mori, Jun Seita, Gerald J Berry, Joseph B Shrager, Ross J Metzger, Christin S Kuo, Norma Neff, Irving L Weissman, Stephen R Quake, and Mark A Krasnow. A molecular cell atlas of the human lung from single cell RNA sequencing. August 2019.
- [2] Minoru Kanehisa, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.*, 44(D1):D457–62, January 2016.
- [3] F Alexander Wolf. Alex wolf - Blog/171223\_AnnData\_indexing\_views\_HDF5-backing. [https://falexwolf.de/blog/171223\\_AnnData\\_indexing\\_views\\_HDF5-backing/](https://falexwolf.de/blog/171223_AnnData_indexing_views_HDF5-backing/). Accessed: 2020-6-10.

- [4] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: Simulation of single-cell rna sequencing data. *Genome Biology*, page 174, September 2017.
- [5] Jose Ordovas-Montanes, Daniel F Dwyer, Sarah K Nyquist, Kathleen M Buchheit, Marko Vukovic, Chaarushena Deb, Marc H Wadsworth, 2nd, Travis K Hughes, Samuel W Kazer, Eri Yoshimoto, Katherine N Cahill, Neil Bhattacharyya, Howard R Katz, Bonnie Berger, Tanya M Laidlaw, Joshua A Boyce, Nora A Barrett, and Alex K Shalek. Allergic inflammatory memory in human respiratory epithelial progenitor cells. *Nature*, 560(7720):649–654, August 2018.
- [6] M D Luecken, M Büttner, K Chaichoompu, A Danese, M Interlandi, M F Mueller, D C Strobl, L Zappia, M Dugas, M Colomé-Tatché, and F J Theis. Benchmarking atlas-level data integration in single-cell genomics. May 2020.
- [7] Mingfeng Liao, Yang Liu, Jing Yuan, Yanling Wen, Gang Xu, Juanjuan Zhao, Lin Cheng, Jinxiu Li, Xin Wang, Fuxiang Wang, Lei Liu, Ido Amit, Shuye Zhang, and Zheng Zhang. Single-cell landscape of bronchoalveolar immune cells in patients with covid-19. *Nature Medicine*, page 174, May 2020.
- [8] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, OSDI’16*, pages 265–283, USA, November 2016. USENIX Association.
- [9] F Chollet. keras. 2015.
- [10] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.*, 19(1):15, February 2018.
- [11] C S Smillie, M Biton, J Ordovas-Montanes, K M Sullivan, and others. Intra-and inter-cellular rewiring of the human colon during ulcerative colitis. *Cell*, 2019.
- [12] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. May 2015.
- [13] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (CMD) for Domain-Invariant representation learning. February 2017.
- [14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv [cs.LG]*, December 2014.
- [15] MD Luecken, M Büttner, K Chaichoompu, A Danese, M Interlandi, MF Mueller, DC Strobl, L Zappia, M Dugas, M Colomé-Tatché, and FJ Theis. Benchmarking atlas-level data integration in single-cell genomics. *bioRxiv*, 2020.
- [16] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12):1053–1058, December 2018.
- [17] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.
- [18] Bonnie Berger Brian Hie, Bryan Bryson. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Naure Biotechnology*, 37(June):685–691, 2019.
- [19] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-Ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nat. Methods*, 16(12):1289–1296, December 2019.

- [20] Maren Büttner, Zhichao Miao, F Alexander Wolf, Sarah A Teichmann, and Fabian J Theis. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods*, 16(1):43–49, January 2019.
- [21] Peter Rousseeuw. Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *comput. appl. math.* 20, 53-65. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987.
- [22] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.
- [23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [24] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.
- [25] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *ArXiv*, abs/1703.01365, 2017.
- [26] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *ArXiv*, abs/1706.03825, 2017.



# Insi2vec: A framework for inferring from single-cell and spatial multi-omics

  
US 20220180975A1

(19) **United States**  
(12) **Patent Application Publication** (10) **Pub. No.: US 2022/0180975 A1**  
**Regev et al.** (43) **Pub. Date: Jun. 9, 2022**

(54) **METHODS AND SYSTEMS FOR DETERMINING GENE EXPRESSION PROFILES AND CELL IDENTITIES FROM MULTI-OMIC IMAGING DATA**

(71) Applicants: **The Broad Institute, Inc.**, Cambridge, MA (US); **Massachusetts Institute of Technology**, Cambridge, MA (US)

(72) Inventors: **Aviv Regev**, Cambridge, MA (US); **Eeshit Dhaval Vaishnav**, Cambridge, MA (US)

(21) Appl. No.: **17/553,691**

(22) Filed: **Dec. 16, 2021**

**Related U.S. Application Data**

(63) Continuation-in-part of application No. 17/426,453, filed on Jul. 28, 2021, filed as application No. PCT/US2020/015481 on Jan. 28, 2020.

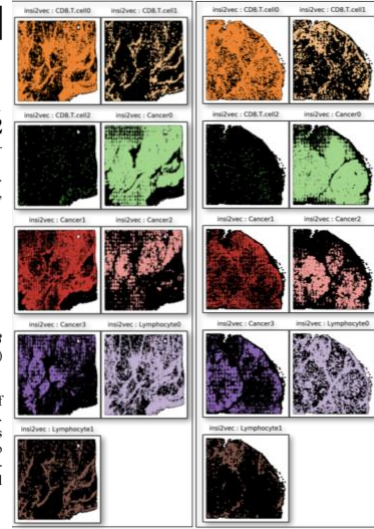
(60) Provisional application No. 62/811,528, filed on Feb. 27, 2019, provisional application No. 62/797,831, filed on Jan. 28, 2019.

**Publication Classification**

(51) **Int. Cl.**  
**G16B 40/30** (2006.01)  
**G16B 25/10** (2006.01)  
**G06N 3/08** (2006.01)

(52) **U.S. Cl.**  
CPC ..... **G16B 40/30** (2019.02); **G06N 3/08** (2013.01); **G16B 25/10** (2019.02)

(57) **ABSTRACT**  
The present disclosure relates to systems and method of determining transcriptomic profile from omics imaging data. The systems and methods train machine learning methods with intrinsic and extrinsic features of a cell and/or tissue to define transcriptomic profiles of the cell and/or tissue. Applicants utilize a convolutional autoencoder to define cell subtypes from images of the cells.  
**Specification includes a Sequence Listing.**



This patent application describes *insi2vec*, *A framework for inferring from single-cell and spatial multi-omics* ([U.S. Patent Application No. 17/553,691](https://patents.google.com/patent/US20220180975A1/en)): *Methods and systems for determining gene expression profiles and cell identities from multi-omic imaging data.* **Contribution:** Co-inventor, with Prof. Aviv Regev.

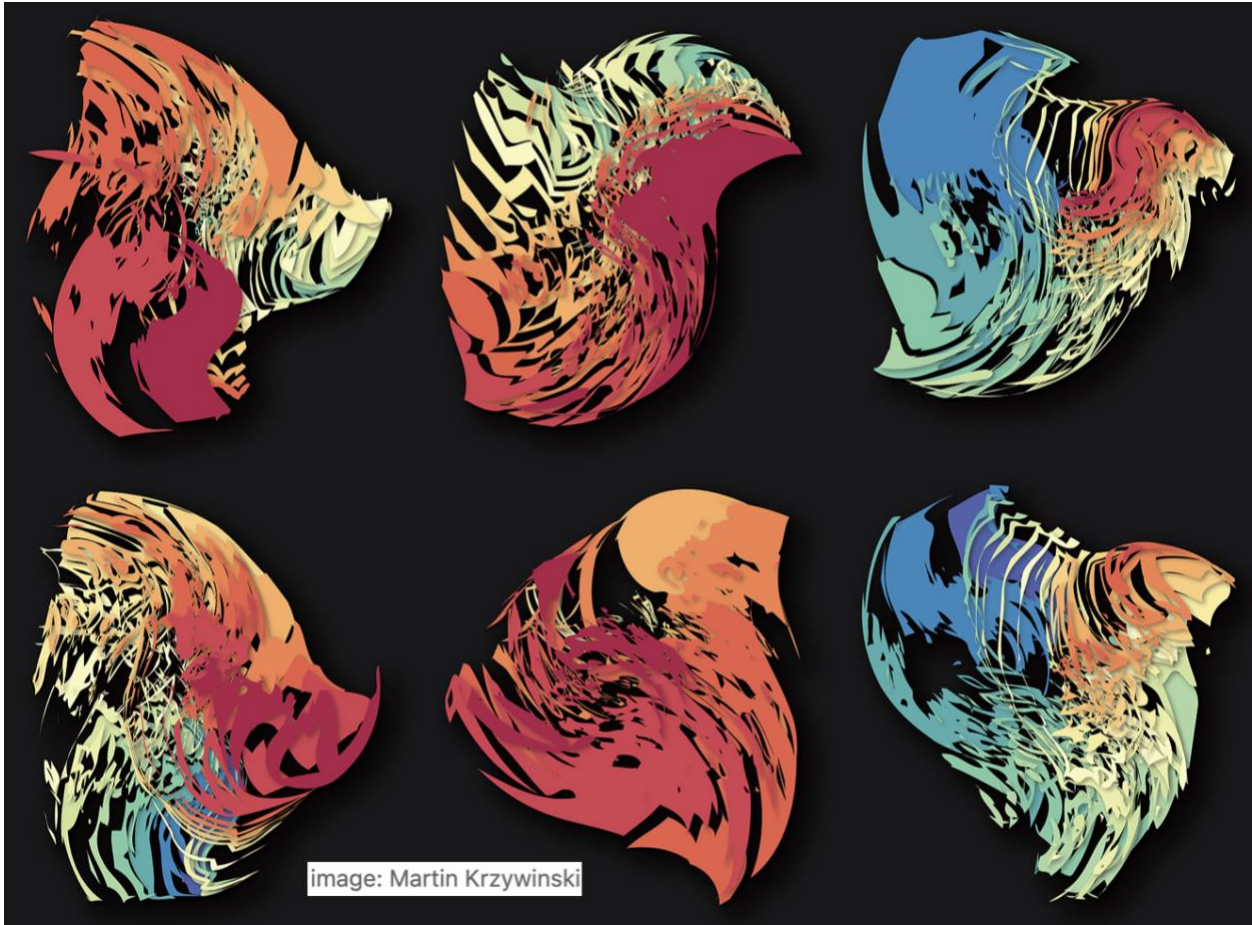
The patent application for *insi2vec*, describes: (i) a spatio-transcriptomic definition of cell identity using cell intrinsic and cell extrinsic features, and methods for predicting spatial gene expression patterns from (ii) single-cell RNA-sequencing measurements and (iii) histology.

The patent application can be accessed at:

<https://patents.google.com/patent/US20220180975A1/en>.



## Discussion



The thesis described our progress towards building ‘foundation models’ for life science.

In the first part of this thesis, I described a framework for thinking about sequence→function questions. The sequence→function relationship is fascinating. Over the (relatively shorter) time scales that humans tend to think about, sequence→function appears to be the direction of causal flow. However, there is also evidence that the inverse relationship may form a causal loop<sup>1-2</sup>. I like to call this phenomenon ‘mutation-selection entanglement’. There may be more to find here.

Designing a biological sequence with a desired function is fundamental to genome medicine, and over the next decade (and beyond), an enormous class of therapeutic programs and target

discovery strategies will involve the design of (and inference from) biological sequences. A summary of our insights from working towards this goal:

**(i) Robustness and Evolvability.** Not only can we engineer sequences to have the desirable function (like expression levels), we have a framework for engineering them to be robust/evolvable. We think this can be a huge advantage in therapeutics (*eg*: for addressing the short-lived nature of existing sequence-based therapies) and in synthetic biology. As Bianco *et al.*, write in their generous News and Views article<sup>3</sup> about our work: "*...the potential for bioengineering and cellular engineering is enormous. In fact, testing for evolvability can open the door to more stable industrial bioengineering pipelines, as just one example. Mutational robustness shapes the fitness landscape so that fitness peaks are tall but wide instead of narrow, indicating that a broader set of mutations is now buffered and can be tolerated without significant loss of fitness. Selecting for a robust system opens the door, in my opinion, to more efficient production pipelines.*"

**(ii) Modeling.** The accuracy of our sequence→function model was enabled by our unique outlook on modeling the sequence→function relationship: We aren't building a model for simply representing the sequence space. On the contrary, we are modeling the interactions of a sequence with its environment that lead to the function we focus on (e.g.: how a promoter sequence interacts with its environment to generate expression). This is a fundamentally different way of looking at model building, compared to approaches that involve the use of large language model analogs for biological sequence representations where the goal is to model distributions within the sequence space itself. In contrast, our goal isn't just to learn a probability distribution in the

sequence space, without a principled consideration of how these sequences interact with their environment and lead to their function.

**(iii) *Measurement.*** Our approach towards model building is only possible because of the scale and quality of experimental data we have been able to generate. The experimental approaches in our work focus on the measurement of function corresponding to large random samples from the sequence space: in sharp contrast to approaches that focus on mutating/perturbing existing biological systems. For learning accurate sequence→function models, local neighborhoods of existing natural sequences are suboptimal; and our approach of training on de-novo random samples of the sequence space performs significantly better. Additionally, random sequences are exponentially cheaper to synthesize at massive scales (compared to defined sequences), enabling high throughput experimental generation of large-scale labelled sequence-function pair datasets.

**(iv) *Generalizability.*** As long as one operates within the constraints of a system where precise, large-scale sequence→function measurements for random samples from the sequence space are possible, our framework is generalizable across an enormous class of sequence design problems that involve DNA/RNA/peptides. As Bianco *et al* write in their article<sup>3</sup> about our work, *"...Basically, the model learns the space of possible solutions and it is capable of generalizing to a new set of solutions. This is of paramount importance because expression engineering is a fundamental part of industrial bioengineering, especially metabolic engineering. But it is even more important because it allows for uncovering the whole complexity of the organism fitness landscape, which can now be computationally revealed on call."*

Moreover, as Wagner *et al* note in their kind *Nature News & Views* story<sup>4</sup> about our paper, "...And, notably, like other applications of deep learning used in the past few years in biology, such as the development of a tool to predict protein folding [Alphafold], it will enable scientists to answer a broader spectrum of questions than any one group of authors could possibly address."

In this thesis, I also describe the Expression Conservation Coefficient (ECC), which helps detect selection pressures from population genetics data. Existing approaches for identifying signatures of selection on regulatory regions rely on disparate assumptions and data types, and thus (to the best of our knowledge) there currently do not exist appropriate per-gene selection measurements against which we can directly compare the ECC. Below, we contrast the existing body of work in this area, with ECC:

*TF binding (or motif) centric approaches.* One pioneering approach<sup>5</sup> uses mutations within motif-predicted TF binding sites to identify signals of conservation (in *Drosophila*), but without regard for how different TFs impact expression or how TFs interact (this was not possible at the time). This approach was applied to TF(s)-enhancer(s) pairs where the TF was known to play an important role in regulation of the enhancer(s). A similar, more recent study<sup>6</sup> created affinity models based on ChIP-seq data for each TF (in human) and used these models to infer selection from the predicted perturbation of binding. It is unclear how one would generalize such approaches to integrate across all TFs and all regulatory regions, and how their results would compare to the ECC, because each TF would provide a different answer. Earlier approaches<sup>7-8</sup> did not always have TF binding data, but used motifs as surrogates. Nevertheless, the approaches were similar in

principle, in that all of them focus on motifs or TFs but do not integrate across the regulatory sequence. This is a key difference from the ECC because, as we have shown in previously<sup>9</sup>, and others have predicted<sup>10</sup>, strong scoring motifs only account for a portion of a gene's regulation.

*Reporter based approaches.* Other methods<sup>11</sup> use the measured effects of mutations within regulatory regions assayed in a reporter assay (in human). Consequently, these can only be applied to regulatory regions for which mutation effects have been experimentally determined, which is not available for the vast majority of sequences one would encounter.

*Mutation counting based approaches.* Methods that are based on counting mutations within a regulatory region (or inter-species comparisons of regulatory sequence) often require a background set of sequences that are assumed to be neutral against which to compare<sup>12-15</sup>. However, in a genome as compact as *S. cerevisiae*'s, it is not clear if there are sufficient locations that are non-functional. Furthermore, these methods assume that the mutation rate is uniform across the genome, which is unlikely to be the case. Finally, we did show that sequence divergence within promoter sequences (*i.e.*, the numerator in such methods) does not correlate with other measures of selection (**Extended Data Fig. 4c**), arguing that such methods would not fare well in a comparison with the ECC.

In the second part of this thesis, I proposed frameworks for thinking about gene expression. Expression lends itself beautifully to the study of genotype→phenotype→fitness. If there is a phenotype that is better suited to this line of scientific inquiry than gene expression, I haven't

found it yet! This part of the thesis focused on applications of these frameworks to cancer, brain and COVID-19 research.

With ATLAS, we introduced a novel framework for predicting and prioritizing putative human disease drug targets from single-cell gene expression measurements. As with all hypothesis generating machine learning methods, one must be extremely careful when interpreting these results and treating them as definitive. For instance, the COVID-19 REP dataset that we used for one of the validations shown above was generated using Vero E6 cells, which are monkey kidney epithelial cell lines. These in-vitro results may not necessarily translate to complex real world biological systems. Even though our intersections with drug targets found from other independent sources is a promising and interesting result, these results must be looked at with extreme caution and require comprehensive further validation using experimental assays and multi-stage rigorous clinical trials before entering the clinic. It is encouraging, though, that our results align with observations made from completely independent ways of analyzing COVID-19 in the context of putative drug targets from scRNA-seq, PPI and drug re-purposing studies. Protein and RNA levels are often known to be uncorrelated in biological systems, and the fact that our results hold across these domains is quite promising. We would also like to emphasize that while the results from our approach, like from any hypothesis generating approach, may provide very interesting and promising putative drug targets, it is imperative that these be validated experimentally and put through rigorous rounds of evaluation. This is an essential step to before any of these results may be considered for real world applications.



We expect utility of feature importance metrics, such as SHAP values, for non-linear and black-box models, in order to identify and characterize complex biological phenomena, to broadly impact the life sciences. We hope that the early demonstration of results using these simple models and variable importance metrics here spurs exponential developments in better approaches for studying a wide range of diseases and biological phenomena. As single cell data is being generated at an ever-faster pace, traditional methods face massive challenges with this unprecedented scale of data. ATLAS has thus been developed with an emphasis on scalability and efficiency.

In summary, I see neural networks' (and more generally, machine learning's) important role in the design→build→test→learn cycle as a starting point. There are innumerable intriguing open questions, downstream of (and orthogonal to) this cycle. Inductive application of principles over (evolutionary) time and (sequence) space, as introduced here<sup>16</sup>, will lead us to interesting answers to some of these questions.

Thank you for reading!

## References

1. Monroe, J.G., Srikant, T., Carbonell-Bejerano, P. *et al.* Mutation bias reflects natural selection in *Arabidopsis thaliana*. *Nature* **602**, 101–105 (2022).
2. Burgess, D.J. Tuning mutagenesis by functional outcome. *Nat Rev Genet* **23**, 135 (2022).
3. Bianco, S. Artificial Intelligence: Bioengineers' Ultimate Best Friend. *GEN Biotechnology*. Apr 2022. 140-141.
4. Wagner. A. AI predicts the effectiveness and evolution of gene promoter sequences. *Nature News and Views*. March 2022.
5. Moses, Alan M. 2009. “Statistical Tests for Natural Selection on Regulatory Regions Based on the Strength of Transcription Factor Binding Sites.” *BMC Evolutionary Biology* 9 (December): 286.
6. Liu, Jialin, and Marc Robinson-Rechavi. 2020. “Robust Inference of Positive Selection on Regulatory Sequences in the Human Brain.” *Science Advances* 6 (48).
7. Gasch, Audrey P., Alan M. Moses, Derek Y. Chiang, Hunter B. Fraser, Mark Berardini, and Michael B. Eisen. 2004. “Conservation and Evolution of Cis-Regulatory Systems in Ascomycete Fungi.” *PLoS Biology* 2 (12): e398.
8. Habib, Naomi, Ilan Wapinski, Hanah Margalit, Aviv Regev, and Nir Friedman. 2012. “A Functional Selection Model Explains Evolutionary Robustness despite Plasticity in Regulatory Networks.” *Molecular Systems Biology* 8: 619.
9. Boer, Carl G. de, Eeshit Dhaval Vaishnav, Ronen Sadeh, Esteban Luis Abeyta, Nir Friedman, and Aviv Regev. 2020. “Deciphering Eukaryotic Gene-Regulatory Logic with 100 Million Random Promoters.” *Nature Biotechnology* 38 (1): 56–65.
10. Tanay, Amos. 2006. “Extensive Low-Affinity Transcriptional Interactions in the Yeast Genome.” *Genome Research* 16 (8): 962–72.
11. Smith, Justin D., Kimberly F. McManus, and Hunter B. Fraser. 2013. “A Novel Test for Selection on Cis-Regulatory Elements Reveals Positive and Negative Selection Acting on Mammalian Transcriptional Enhancers.” *Molecular Biology and Evolution* 30 (11): 2509–18.
12. Haygood, Ralph, Olivier Fedrigo, Brian Hanson, Ken-Daigoro Yokoyama, and Gregory A. Wray. 2007. “Promoter Regions of Many Neural- and Nutrition-Related Genes Have Experienced Positive Selection during Human Evolution.” *Nature Genetics* 39 (9): 1140–44.
13. McDonald, J. H., and M. Kreitman. 1991. “Adaptive Protein Evolution at the Adh Locus in *Drosophila*.” *Nature* 351 (6328): 652–54.
14. Andolfatto, Peter. 2005. “Adaptive Evolution of Non-Coding DNA in *Drosophila*.” *Nature* 437 (7062): 1149–52.
15. Hahn, Matthew W. 2007. “Detecting Natural Selection on Cis-Regulatory DNA.” *Genetica* 129 (1): 7–18.
16. Vaishnav, E.D., de Boer, C.G., Molinet, J. *et al.* The evolution, evolvability and engineering of gene regulatory DNA. *Nature* **603**, 455–463 (2022).