NORTHWESTERN UNIVERSITY

Discovering Regulatory Insights from Gene Expression Dynamics

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Interdisciplinary Biological Sciences

By

Justin D. Finkle

EVANSTON, ILLINOIS

June 2019

ABSTRACT

Discovering Regulatory Insights from Gene Expression Dynamics

Justin D. Finkle

Cells are complex, autonomous machines that integrate many environmental cues to execute a desired response. Though this property makes cells versatile, it presents significant design challenges when, to treat diseases, we must alter cellular responses. To understand changes to the complex regulatory pathways that cause diseases, studies often investigate the differential gene expression between genetically or chemically differing cell populations. This approach transformed the discovery of genetic drivers of disease and possible therapies. Current high-throughput technologies also provide a wealth of time-series data that captures complex regulatory dynamics, yet many current analyses do not capitalize on this temporal information to provide quantitative predictions of gene expression in untested conditions. A better understanding of gene expression dynamics will lead to more detailed and quantitative models of cellular regulation. This improvement can accelerate our understanding of biological systems, guide future experiments, and enhance our ability to control cellular behavior.

In the following work, I present two distinct approaches that utilize gene expression dynamics to elucidate systems that regulate transcription. I developed algorithms to

identify genes that regulate each others' expression, create dynamical systems models of expression that accurately predict gene expression in multiple contexts, and gain insight into the regulators of specific transcriptional responses. I validated my algorithms on *in silico* and *in vitro* data. I demonstrate how these techniques revealed unique insights into transcriptional regulation by PI3K and Sprouty. My work illustrates how the principled use of temporal information can improve our understanding of biological systems, and I hope it encourages others to collect more time-series data in the future.

Acknowledgements

This work would not be possible without the incredible support of many individuals. My mentor, Neda Bagheri, was incredibly patient and supportive throughout this process. Anyone who has met her, or heard her laughter emanating from her office, knows that she brings joy and optimism to the scientific process. Thank you Neda, for helping me develop as a scientist, presenter, and mentor.

Other than my brief foray into pipetting during a rotation in the Tyo lab, my work relied on the excellent support of my collaborators. Thank you to Behnam Nabet and Jon Licht, who were kind enough to trust me with their data, and always accessible to help answer my naive questions. Thank you to Jia Wu, who hates praise, but is a brilliant scientist, passionate coder, and my favorite co-first author.

I would like to thank my committee members for their valuable feedback and support: Ravi Allada, Curt Horvath, Bill Kath, and Josh Leonard. Thank you to the members of the IBiS and ChBE offices, who helped me in my continuous struggle to figure out which department got to process my paperwork. Additional thanks to Stephanie Brehm for helping me navigate other administrative hurdles, and for her friendship.

Whether they know or not, I was informally mentored by three outstanding people at Northwestern. Matt Moura was forced to take me under his wing, and he helped me learn Python the hard way. To be honest, I learned more Python from Dante Pertusi and James Jeffryes; Matt won't be upset with me for saying so. Mark Ciaccio taught me

the basics of genomics experiments and machine learning. Adam Hockenberry challenged me to think differently about diverse problems and provided stimulating conversations on every topic. Thank you all for shaping my views of research and programming.

I also want to thank all former and current members of the Bagheri Lab family. I enjoyed critical discussion of science and nonsense with Sebastian Bernasek, Josh Levy, Aaron Oppenheimer, Albert Xue, Kasim Fassia and Joe Muldoon. All visualization credits go to Jessica Yu. Thank you all for making the Bagheri lab a memorable place. Incredible thanks to my other friends near and far, who enriched my non-academic life.

My family knows nothing about what I do, yet they still provide endless support. Thank you to Uppie and Nana, who, at over eighty years old, make the most effort of any family members to understand my, and even Carissa's, research. When everything else fails, their pride in me remains. I owe everything to my parents, Lauren and Robert. Though neither is a scientist, they taught me how to be one. From you, I learned to navigate the world with curiosity and excitement.

Finally, thank you to the two special ladies in my life. One constantly notifies me she is hungry and the other is a dog. River was at my side through countless hours of typing. She keeps me moving and reminds me—even if I will never ever catch them—to always chase my passions like a bunny. This thesis is obviously dedicated to Carissa. With you, every challenge or chore is made fun. You make me want to grow, even when I don't want to get out of bed. I can't wait to explore the world with you.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

Tremendous efforts to characterize cellular components and signaling pathways yielded remarkable biological insights and areas to improve therapies. The impact of these efforts is evident in cancer biology, where progress in targeted therapies—and now immunotherapies—significantly improves patient outcomes[1,2]. However, in many cases, cellular complexity allows cancers to develop resistance to the treatment, or causes serious side effects[3–8]. To overcome these limitations and design effective control strategies, we first need to characterize the cellular systems we aim to control. High-throughput technologies quantify properties of DNA, RNA, proteins, and metabolism, and greatly advance our global understanding of cells[9–12].

Though proteins are often the molecule of interest to researchers studying cellular processes, the ability to measure RNA levels genome-wide has historically out-paced that of proteins. And, while techniques to measure protein-DNA interactions and chromatin accessibility yield powerful information about DNA states, it remains challenging to determine downstream effects of those states[10,13,14]. The measurement of gene expression provides an easily measured middle ground and therefore remains a key proxy to infer how cells respond to their environment[9,15–17].

To understand the underlying causes of diseases, differential gene expression studies often compare RNA levels between genetically or chemically differing cell populations.

This approach identifies global changes in transcription and enables the inference of functional roles of the applied perturbations[15,18–20]. By testing if the differentially expressed genes share similar functional annotations or overlapping pathways, the results can be translated into biological meaning[21–24].

However, these qualitative associations provide limited insight into how the system will respond in a different environmental context or what specific components will mediate that response. This lack of predictive capability is the motivation behind a large portion of the work in the following thesis. It surprised me that we could gather so much data, yet come to conclusions as non-specific as "mRNAs overexpressed...are involved in cell respiration"[25], with no indication of how those genes became overexpressed or how they would respond in different conditions. Retrospectively, the idea that a single genomics experiment could supply enough information to infer concrete biological mechanisms from the near infinite space of complex possibilities is, perhaps, a bit naive. The available data limits the type of models we can create, and the model subsequently limits the insights we can gain[26,27].

For the remainder of this introductory chapter, I will discuss the trade-offs of different classes of mathematical models. Chapters 2-4 represent my attempts at using time-series gene expression data to gain the benefits of multiple types of mathematical models and thereby produce more quantitative and easily tested biological hypotheses. I validate the accuracy of each developed algorithm and highlight how each result increases our understanding of the biological system analyzed.

1.1. Mathematical modeling of biological systems

With the advent of sequencing technologies, a deluge of biological data has become available on an array of databases[28–31]. Processing this volume of information alone requires

mathematical and computational approaches[32,33], but mathematical models also provide a critical role in understanding biological systems. Models help provide a lens through which disparate data can be integrated and interpreted. In biology, a mathematical model also describes how we believe a physical process occurs, which we can use to predict what will happen in an untested situation. When the results of a test do not match the prediction, the model can be updated[34].

There are many types of mathematical models that encompass different levels of detail and complexity[27,35–37]. At one end of the spectrum, statistical methods—such as statistical hypothesis tests and linear regression—fit a model to data with minimal prior assumptions. On the other end, systems of differential equations often specify detailed kinetic interactions between species in the model[26,27]. Here I discuss the trade-offs of using gene expression data with each type of model.

1.1.1. Statistical models are applicable to many situations

Statistical methods are necessary to quantify gene expression[19,38,39]. They are also instrumental for the analysis of expression data, such clustering genes with similar values[40], comparing expression samples[41–43], identifying differentially expressed genes[44,45], and inferring regulatory networks[46–48]. To gain biological insight from gene expression, studies frequently assess whether changes in gene expression are significantly associated with Gene Ontology (GO) terms[21,24]—which curate proven biological functions with genes—or with specific pathways[22]. Both of these approaches rely on statistical methods[49,50], and are used to interpret gene expression in many studies.

However, a major limitation of these approaches is that it falls on the researcher to infer the significant biological insight from expression analysis. In my experience, the

results from these tests are often vague and abstract. I believe this ambiguity leads to a "choose your own adventure", in which researchers focus on the results that best fit their narrative. Additionally, the acquired biological insights are mostly qualitative, which still leaves the researcher to hypothesize how gene expression will vary in an untested context.

In contrasts, supervised machine learning approaches build models that map input data to an output; the are widely used to uncover regulation in biological systems[17,26,27,40,46,51,52]. These methods are mostly agnostic to the input data, and a model is trained regardless of the appropriateness or quality of the data. Many of these methods find linearly independent relationships between input and output variables, and they can therefore assess the influence of multiple input variables on the output[53–55]. These attributes make supervised learning approaches ideal to infer gene regulatory networks (GRNs) from gene expression data. GRNs describe how genes regulate one another by representing genes as nodes and their interactions as edges in a network[27,47,51].

This feature makes supervised methods easily generalized to many problems, but they have some disadvantages when applied to gene expression data. The models assume a relationship between the input and output variables (*e.g.* linear), and the researcher must infer the physical mechanism that could result in such a relationship. New input data is also needed to predict the output response. For example, in the simplest case of linear regression, if gene y linearly depends on gene x, to predict the expression of gene y, we must know the expression of gene x. In the case of an inferred GRN, the prediction itself is likely impossible, due to the inter-connectivity of the inputs and outputs. Thus, the statistical models are only used to infer the structure of the GRN, leading to the second problem. Finally, because edges in a GRN are statistically inferred, there is no guarantee that an edge represents a direct interaction between the two genes[47,56].

In the following chapters of my thesis I demonstrate how analyzing time-series gene expression data can overcome some of these limitations. In Chapter 2, I present work that uses time-series data to more accurately infer the structure of GRNs. In Chapter 3, I develop and validate an algorithm, termed **Diff**erential **Ex**pression in **Py**thon (DiffExPy), that uses time-series data to identify when the expression of genes is regulated and by which transcription factors. In both Chapters 3 and 4, I demonstrate how DiffExPy yields new, testable biological insights into regulation by PI3K and Sprouty, respectively.

1.2. Time-series data provides a bridge between models scales

A limitation of statistical methods is that they are ill-suited to created quantitative predictions of gene expression in different contexts. Systems of differential equations are detailed mathematical models that can provide this type of prediction. Differential equations are often used to relate the quantity of a physical entity with factors that govern its rate of change. Thus, systems of differential equations are used to describe how biological species, such as genes, proteins or metabolites, impact each others' rates of change[57–60]. Stochastic variants also exist that capture heterogeneity and noise within gene expression systems[61,62]. Because systems of differential equations capture kinetic relationships between species, they can be used in multiple contexts and updated when their predictions do not match newly observed data[57]. These features make differential equation models helpful for testing strategies to control the response of a chemical/biological system.

Though systems of differential equations are powerful models, creating a system that predicts genome-wide expression is challenging[63]. Accurately fitting parameters of differential equations for all of the genome requires much more data than can typically be measured, even with high-throughput technologies. Therefore, differential equation models only exist for a few, well-studied systems[64–67]. Using a defined basis set of interaction

types genetic and sparse regression algorithms can generate differential equation models directly from data. However, current gene expression technologies cannot produce the highly sampled, low-noise data these algorithms require[68–71].

I developed DiffExPy to bridge this gap. In Chapter 3, I describe how DiffExPy fits ensembles of stochastic differential equations independently for many genes using time-series gene expression data. Using previously published RNA-seq data[25], I train the models and validate their predictions. In both Chapters 3 and 4, I demonstrate how the DiffExPy models provide novel insights into the transcriptional regulation and make quantitative predictions. Overall, the models generated by DiffExPy represent a first step in a data-driven approach toward creating differential equation models for hundreds to thousands of genes. I expect them to provide a basis for developing more detailed models that improve our understanding and control of biological systems.

CHAPTER 2

Windowed Granger causal inference strategy improves discovery of gene regulatory networks

This work was published with Jia J. Wu (co-first author), and Neda Bagheri in the Proceedings of the National Academy of Sciences of the United States of America, 2018[48].

2.1. Abstract

Accurate inference of regulatory networks from experimental data facilitates the rapid characterization and understanding of biological systems. High-throughput technologies can provide a wealth of time-series data to better interrogate the complex regulatory dynamics inherent to organisms, but many network inference strategies do not effectively use temporal information. We address this limitation by introducing Sliding Window Inference for Network Generation (SWING), a generalized framework that incorporates multivariate Granger causality to infer network structure from time-series data. SWING moves beyond existing Granger methods by generating windowed models that simultaneously evaluate multiple upstream regulators at several potential time delays. We demonstrate that SWING elucidates network structure with greater accuracy in both *in silico* and experimentally-validated *in vitro* systems. We estimate the apparent time delays present in each system and demonstrate that SWING infers time-delayed, gene-gene interactions that are distinct from baseline methods. By providing a temporal framework to infer the underlying directed network topology, SWING generates testable hypotheses for novel gene-gene influences.

2.2. Introduction

Elucidating gene-gene regulation is a fundamental challenge in molecular biology, and high-throughput technologies continue to provide insight about the underlying organization, or topology, of these interactions. Accurate network models representing genes (nodes) and regulatory interactions (edges) infer information from many observed heterogeneous components while minimizing the effects of noise and hidden nodes. Many methods infer gene regulatory networks (GRNs) from expression profiles[27], but each suffers from limitations—assumptions of linearity, univariate comparisons, or computational complexity—and most ignore temporal information in time-series data. Understanding the temporal dynamics of gene/protein expression is critical to elucidating responses involved in cell cycle, circadian rhythms, DNA damage, and development[72–75].

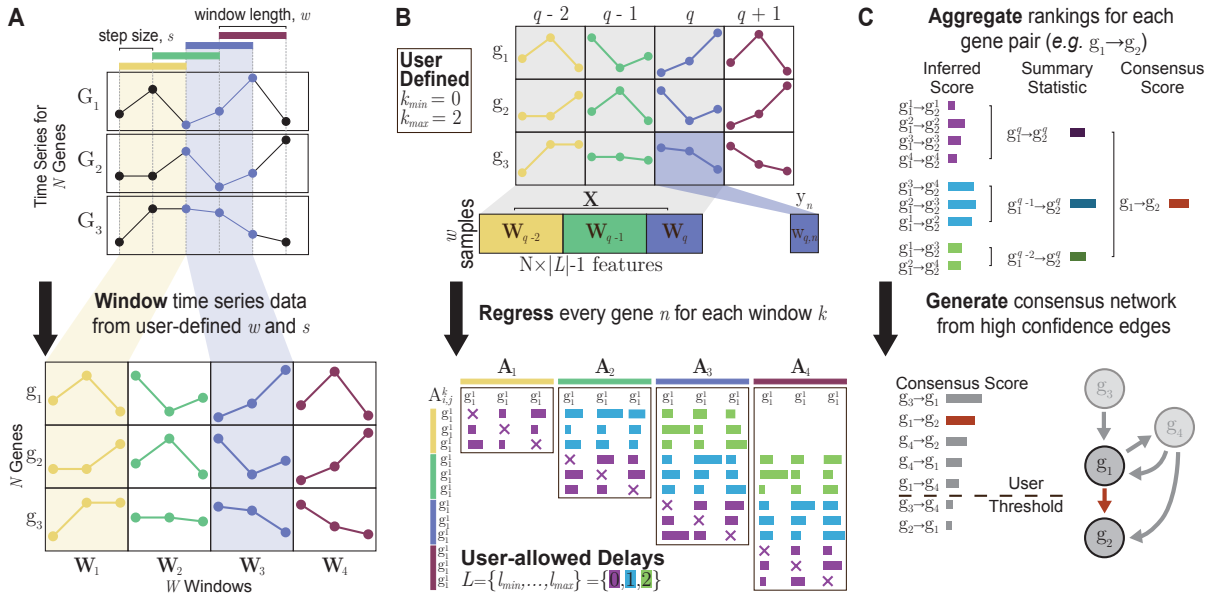


Figure 2.1. Overview of the SWING framework. (A) Time-series data is divided into windows with a user-specified width, w. (B) For each window, inference is performed by iteratively selecting response and explanatory genes. The subset of available explanatory genes is defined by the minimum and maximum user-allowed time delays. (C) Edges from each window model are aggregated into a single network representation of the biological interactions between measured variables.

Existing methods to infer GRNs from time-series expression profiles include dynamical models, statistical approaches, and hybrids of the two[27,76–78]. Dynamical systems models of differential equations can forecast future system behaviors and characterize formal properties such as stability[79], but these models are computationally intractable for large GRNs due to extensive and explicit parameterization requirements[80]. Statistical inference methods—such as regression schemes, mutual information, decision trees, and Bayesian probability[53,55,81]—make no explicit mechanistic assumptions and are often more computationally efficient than dynamical models. However, many implementations of aforementioned algorithms treat time points as independent observations, disregarding time delays associated with transcription, translation, and other processes inherent to

gene regulation[47,54]. Hybrid methods—such as SINDy and Jump3—use statistical methods to optimize the search and parameterization of dynamical models, but they remain computationally expensive and rely on accurate specification of basis functions[82,83].

If the experimental sampling interval is less than or equal to the time delay between a regulator and its downstream target, it is possible to employ Granger causality to incorporate intrinsic delays that are often hidden from measurement[84]. Current implementations of Granger causal network inference methods are limited; the inference (*i*) is conducted pairwise, prohibiting simultaneous assessment of multiple upstream regulators, (*ii*) has a single user-defined delay, which assumes a uniform delay between all regulators and their targets, or (*iii*) requires each explanatory variable, assessed at multiple delays, to be selected as a group[85–89]. Thus, their implementation has limited broad utility in biological systems with heterogeneous time delays.

To allow for multiple time delays to affect downstream target nodes, we introduce an extensible framework to infer GRNs from time-series data, termed Sliding Window Inference for Network Generation (SWING). SWING embeds existing multivariate methods, both linear and nonlinear, into a Granger causal framework that concurrently considers multiple time delays to infer causal regulators for each node. SWING also uses sliding windows to create many sensitive, but noisy, inference models that are aggregated into a more stable and accurate network. We validate the efficacy of SWING on several *in silico* time-series data sets, and existing *in vitro* data sets with corresponding gold standard networks. We show that SWING performs network reconstruction more accurately than baseline methods, and demonstrate that this performance boost is partly attributed to inferring edges that involve an identifiable time delay between upstream regulators and targets. In validation studies analyzing networks derived from *E. coli* and *S. cerevisiae*,

SWING infers networks with distinct topologies, and can therefore be combined with other methods to improve consensus models. The SWING framework is available for use and can be found on GitHub (*https://github.com/bagherilab/SWING*).

2.3. Results

SWING integrates multivariate Granger causality and ensemble learning to infer interactions from gene expression data. First, SWING subdivides time-series data into several temporally-spaced windows based on user-specified parameters (Fig. 2.1A). For each window, edges are inferred from the selected window and previous windows, representing interactions with specific delays. This inference results in a ranked list of time-delayed, gene-gene interactions for each window. (Fig. 2.1B). The ensemble of models is aggregated based on edge rank into a static GRN (Fig. 2.1C). *In silico* and *in vitro* validation confirm notable performance improvements.

2.3.1. SWING improves the inference of *in silico* GRNs

We applied SWING to reconstruct *in silico* GRNs simulated by GeneNetWeaver (GNW)[61]. 20 subnetworks with 10 nodes and non-isomorphic topologies were extracted from *E. coli* and *S. cerevisiae* networks included in GNW to use as gold standards. Networks were inferred from the generated time-series data using existing multivariate methods as a basis for comparison. We employed RandomForest (RF), Least Absolute Shrinkage and Selection Operator (LASSO), and Partial Least Squares Regression (PLSR)[51,53,55], which represent the areas of sparse, nonlinear, and PLS-based regression. We implemented the SWING chassis and compared the performance of each SWING frontline method with its base method: SWING-RF vs. RF, SWING-LASSO vs. LASSO, SWING-PLSR vs. PLSR.

To capture short-term dynamics consistent with simulated perturbations, we set the window size to roughly half the duration of the time series. The minimum and maximum lags were set to $k_{min} = 1$ and $k_{max} = 3$, which correspond to 50 and 100min. We compared the group of inferred networks by calculating the mean increase in the area under the precision-recall (AUPR) and area under the receiver operating characteristic (AUROC) curves of 40 *in silico* networks. Compared to respective baseline methods, SWING shows a statistically significant increase in AUROC and AUPR for many of the 10-node networks (Fig. 2.2A and SI Appendix, Table 2.S1) and across all of the 100-node networks (SI Appendix, Fig. 2.S1, Table 2.S1). In particular, RF receives the most notable benefit from SWING; SWING-RF outperforms RF in 39 out of 40 *in silico* networks and application of SWING-RF results in the highest mean AUROC and AUPR for *in silico* networks among tested methods.

2.3.2. SWING infers distinct edges in networks

No single method performs optimally across all data sets, partially due to biases in predicting different network topologies. For example, *E. coli*-derived networks predominately feature fan-out motifs, which RF infers with greater sensitivity. In contrast, *S. cerevisiae*-derived networks contain more cascade motifs, which are inferred with greater sensitivity by linear methods[47].

To determine if SWING methods provide distinct information from RF, LASSO, and PLSR, we ran principal component analysis (PCA) on ranked edge lists predicted by SWING and the corresponding base methods (Fig. 2.2B). We discarded PC1 because it largely explains the overall performance of each inference method (58% variance explained; SI Appendix, Fig. 2.S2). Clustering of results in PC2 and PC3 seems to explain biases

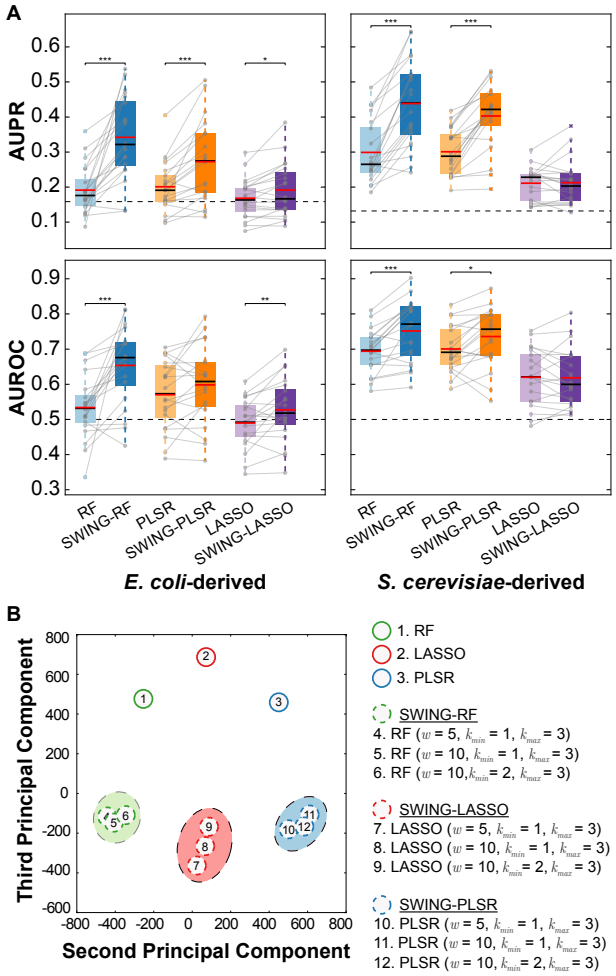


Figure 2.2. SWING improves inference of 10-node *in silico* networks. (A) Changes in AUPR and AUROC in GNW networks. Score changes to individual networks are shown in grey. The mean (red) and median (black) of each score distribution is shown. AUPR and AUROC increase when using SWING-RF or SWING-PLSR compared to their respective base method. SWING-LASSO outperforms LASSO in the *E. coli*-derived networks. The expected score based on random for each metric is shown as a dashed line. n=20 networks, $k_{min} = 1$, $k_{max} = 3$, and $w = 10$ for all networks. p-values were calculated using the Wilcoxon signed-rank test, $^{***}p<0.001$, $^{**}p<0.01$, $^{*}p<0.05$. (B) SWING and non-SWING methods are grouped according to similarity of ranked predictions for 40 10-node *in silico* networks via PCA. PC1 largely separates inference methods based on performance (SI Appendix, Fig. 2.S2), while PC2 separates methods based on underlying base method. Networks inferred by various SWING parameter selections cluster together according to inference type, with SWING methods forming clusters distinct from corresponding base methods.

toward specific network motifs[47]. Along PC2, edge rankings appear to separate based on the internal base method (15% variance explained), while along PC3, SWING edge rankings appear to separate from those of their base methods (5% variance explained). These results suggest that SWING recovers connectivities that are distinct from those recovered from RF, LASSO, and PLSR.

Given that it is difficult to determine *a priori* which methods perform optimally in different contexts, deriving a community network is a good strategy for robustly improving predictions[47]. We evaluated the performance of SWING-Community, which combines SWING-RF, SWING-LASSO, and SWING-PLSR predictions by calculating the mean

rank across all methods for each possible edge. We note that SWING-Community outperforms RF, resulting in a 52% and 8% mean increase in AUPR and AUROC, respectively, suggesting that SWING infers distinct and complementary networks (SI Appendix, Fig. 2.S3).

2.3.3. SWING improves network inference by promoting time-delayed edges

Endogenous reactions, such as protein translation, post-translational modifications, translocation, or oligomerization are often not accounted for in the inference model. However, even if underlying network kinetics are linear (or approximately linear), the resulting dynamics can appear delayed when not all nodes are observed (SI Appendix, Fig. 2.S4A). Delayed behavior in gene expression and protein translation has been established in several studies[90,91].

We estimated the apparent time delay of each interaction in a 10-node GNW network by calculating the pairwise peak cross-correlation between time series of all true regulator and target combinations. The majority of true interactions within GNW networks have a time delay between 0 and 150min (SI Appendix, Fig. 2.S4B). We observe that SWING is more likely to promote edges with an identifiable delay within the range of user-specified parameters (SI Appendix, Fig. 2.S5A). Across all *in silico* networks, SWING-RF promotes 65.8% of true edges with a delay versus 55.4% of true edges without a delay (p=0.018), and SWING-PLSR promotes 67.0% of true edges with a delay versus 47.1% of true edges without a delay (p=6e-6)(SI Appendix, Fig. 2.S5B).

Many of the promoted edges with an identifiable delay are highly ranked by base methods RF and PLSR. In general, delayed true edges ranked in the first quartile by the base method are likely to be promoted, while those ranked lower are no more likely to

be promoted than nondelayed true edges (SI Appendix, Fig. 2.S5B). While SWING is more likely to promote true edges with a delay, the magnitude of this promotion is not consistent across the different base methods or networks. SWING-RF promotes true edges with an apparent time delay by an average of 7.50 ranks relative to true edges without an apparent time delay (p=4.75e-3) for *S. cerevisiae*-derived networks. In contrast, SWING-PLSR promotes true edges with an apparent delay by an average of 7.78 ranks relative to true edges without an apparent time delay (p=6.89e-5) for *E. coli*-derived networks (SI Appendix, Fig. 2.S5B). In one example, *S. cerevisiae* Network 12, SWING-RF improves the AUROC from 0.539 to 0.872, a 61.7% increase relative to the base method. Compared to RF the edge ranking for SWING-RF promotes many true edges, and all of the true edges with a delay are promoted by SWING (SI Appendix, Fig. 2.S6A).

To demonstrate how SWING promotes delayed edges, we highlighted the true edge between Gene 2 (G2) and Gene 1 (G1) in *S. cerevisiae* Network 12. G2 is the only node upstream of G1, and the input data includes an experiment where only G2 is perturbed, thus the delay between G2 stimulation and G1 response is unambiguously isolated (SI Appendix, Fig. 2.S7A). We estimated the delay between G2 and G1 as two time points, or 100min. We shifted the G1 time series by two time points to show that the Pearson correlation of the resulting time series notably increases (SI Appendix, Fig. 2.S6B).

2.3.4. SWING infers apparent time-delayed edges with greater sensitivity in the *E. coli* SOS network

We applied SWING to an *in vitro* 8-node *E. coli* GRN that activates with DNA damage[86,92]. The SOS network contains several complex interactions, including multiple cascades and feedback loops generated by a combination of transcriptional activators and

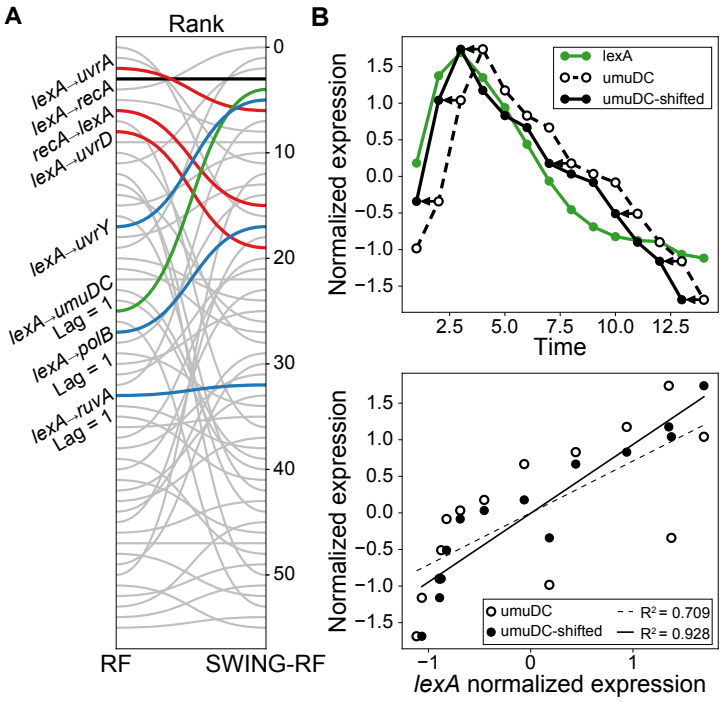


Figure 2.3. SWING promotes edges with apparent time delays and increases correlation between genes. The true network structure is provided in SI Appendix, Fig. 2.S7B. (A) Edge rank comparison for *E. coli* SOS network when using RF and SWING-RF (blue = promoted edges, red = demoted edges, black = no change, grey = false edges, green = *lexA* → *umuDC* analyzed in panels B and C). We report the lag for edges with an apparent time delay. (B) Time series for *lexA* and *umuDC* show better alignment when *umuDC* is shifted by one time period. (C) Improved correlation between *lexA* and *umuDC* the time series of *umuDC* is shifted by one time period.

repressors. We computed the mean of three replicates for each time point following DNA damage inducing Norfloxacin treatment[93].

The sampling strategy for the *in vitro* SOS data is different from that of the *in silico* GNW data. Due to fewer time points, we were restricted to assessing interactions with shorter possible time delays. Using $w = 0.5T = 7$, $k_{min} = 0$, and $k_{max} = 1$, SWING-RF infers the network more accurately than other reported inference algorithms including RF, LASSO, TSNI[93], and BANJO[94]. Because RF is a stochastic method, we ran both RF and SWING-RF 50 times on the SOS network. On average, SWING-RF increases the AUPR from 0.286 to 0.356 (24.6%, p=1.41e-13) and the AUROC from 0.756 to 0.819 (8.3%, p=5.28e-34). To assess promotion of time-delayed edges, we calculated the mean edge ranks across all 50 runs and compared the resulting lists. Though SWING-RF demotes some true edges, it promotes all three edges that exhibit a time delay (Fig. 2.3A). We

highlight the edge between *lexA* and *umuDC* (SI Appendix, Fig. 2.S7B), which has an estimated lag of 6min. When the *umuDC* time series is shifted by this amount the correlation between *lexA* and *umuDC* increases from 0.709 to 0.928 (Fig. 2.3B). These findings reaffirm that SWING improves network inference, in part, by promoting edges with identifiable delays.

2.3.5. SWING accurately infers RegulonDB modules with time-delayed edges

We curated microarray data to infer time-delayed edges from experimentally validated GRNs in *E. coli* (Fig. 2.4A) and *S. cerevisiae* (SI Appendix, Fig. 2.S8). This curated data was aggregated across 18 data sets for *E. coli* and 8 data sets for *S. cerevisiae*, where data was unevenly sampled for time intervals that range from 5 to 120min (SI Appendix, Table 2.S2). To assess the landscape of apparent time delays present in these gene expression data, we performed pairwise cross-correlation lag selection between experimentally-confirmed edges[95]. We reveal that of 2870 experimentally confirmed edges, only 23.7% exhibit an apparent time delay of 0 and 13.7% exhibit a time delay of at least 10min. Surprisingly, only 37.4% of confirmed edges exhibited pairwise correlation (R>0.7, p<1e-5; Fig. 2.4A).

To determine whether lag is associated with modularity and function, we clustered the *E. coli* and *S. cerevisiae* network into smaller modules using MCODE[96] and performed gene ontology enrichment analysis. Several modules, such as those associated with catabolic processes and metal ion binding, are enriched with time-delayed edges of at least 10min (SI Appendix, Tables 2.S3 and 2.S4). Transcription factors are known to regulate genes on a global or combinatorial scale tend to exhibit similar time delays (SI Appendix, Table 2.S5).

To determine if SWING more accurately infers network structure in diverse contexts, we performed cubic spline interpolation to generate evenly sampled time-series gene expression at 10min intervals and benchmarked SWING-Community performance against an ensemble model of RF, LASSO and PLSR (R/L/P) base for each clustered module using this dataset. SWING-Community outperformed R/L/P in subnetworks in which more than 10% of edges are time-delayed (N=12 clusters, 9 clusters with fewer then 10 genes, or fewer than 3 transcription factors were removed from analysis, p=0.031; Fig. 2.4B). As an example, we identified time-delayed properties of key regulators of the *tdcABC E. coli* operon that are responsible for the transport of threonine and serine during anaerobic growth[97]. In particular, our analysis identifies two global transcription factors that bind combinatorially to induce activity in the *tdcABC* operon. *Crp* and *fnr* are global regulators that respond to glucose starvation and anaerobic growth respectively[98,99].

Interestingly, lag analysis identifies 10 and 20min time delays between *crp* and target genes in the *E. coli tdcABC* operon. While the precise delay identified by our analysis is not consistent with that observed in experiments, studies confirm that a delay exists between *crp* induction and the induction of several target genes due to post-translational modification[100,101]. Of 32 edges in the gold standard, SWING identifies 27 true-positive (TP) edges and 5 false-positive (FP) edges (85% TP) while the ensemble model predicts 24 true-positive edges and 8 false-positive edges (75% TP). In this example, SWING-Community infers both time-delayed and non time-delayed edges more sensitively than the R/L/P ensemble model. The false-positive edges inferred by SWING-Community are also within the subset of false-positive edges inferred by the base community method.

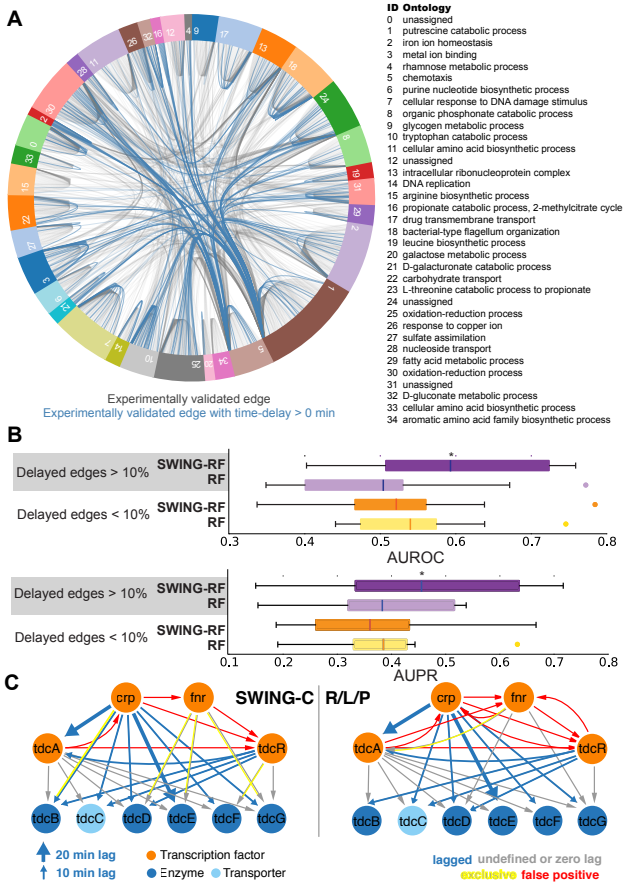


Figure 2.4. Application of SWING on time-delayed gene regulatory network modules in *E. coli*. (A) Circular diagram depicts experimentally validated interactions and gene ontologies present in each module (RegulonDb). Blue edges depict time-delayed interactions inferred using pairwise cross-correlation from curated microarray data. (B) SWING-Community, with $w = 4$, $k_{min} = 1$, $k_{max} = 1$ applied to RegulonDb subnetworks that are and are not enriched with time-delayed edges (fraction of delayed edges is greater than 10%, n=12 subnetworks; fraction of delayed edges is less than 10%, n=14 subnetworks). (C) SWING-Community and RF/LASSO/PLSR (R/L/P) ensemble method applied to *tdcABC* regulon, which is the module found to have the highest enrichment of time-delayed edges (44% edges with a time delay of 10min or greater).

2.3.6. SWING performance is robust across parameters

SWING adds user-defined parameters to baseline methods, which are necessary for window creation and time-delay inference. The selection of these parameters is both context and data specific. We conducted parametric sensitivity analysis of SWING as a function of window size, combinations of k_{min} and k_{max}, and experimental sampling interval

in context of the *in silico* networks and the *E. coli* SOS network (SI Appendix, Figs. 2.S9-2.S14). While SWING outperforms baseline methods over a wide range of window sizes (SI Appendix, Fig. 2.S9), the performance of a single network may differ from other networks, suggesting that the optimal window size is partially dependent on the underlying inference method and network structure. Therefore, user-specified SWING parameters—k_{min}, k_{max}, and w—should be chosen based on the data, and are discussed in detail in Supporting Information: Sensitivity Analysis. Overall SWING outperforms baseline methods for a wide range of possible parameters (SI Appendix, Figs. 2.S9-2.S13).

2.4. Discussion

Tight regulation of gene expression is critical to maintaining robust responses to perturbations and environmental disturbances, and misregulation of intracellular signaling dynamics can lead to a wide variety of diseases. For this reason, uncovering the topology of GRNs is of fundamental interest to the scientific community, since the resulting maps can be used to identify interventions to control cellular phenotypes. Many current methods disregard temporal information and are limited in their ability to accurately infer network topology. Indifference to time delays will be the Achilles heel of many systems biology strategies. We developed a general temporal framework for network inference that accurately uncovers the regulatory structures governing complex biological systems by accounting for these fundamental delays. SWING improves upon existing Granger methods by generating an ensemble of windowed models that simultaneously evaluate multiple upstream regulators at several potential time delays. We validate its utility and performance in several *in silico* (Fig. 2.2A) and *in vitro* (Figs. 2.3 and 2.4B) systems.

2.4.1. Consideration of time delays improves SWING performance and should be integrated in experimental design

Our *in silico* and *in vitro* results demonstrate that promoted edges were enriched for those with apparent time delays (SI Appendix, Fig. 2.S5B), suggesting that network inference is improved, in part, by accounting for temporal information. We support this finding by demonstrating that SWING-RF promotes an edge with a distinct and singular delay (SI Appendix, Fig. 2.S6A). We also used SWING to predict directed edges of several *E. coli* sub-networks using cubic spline interpolated microarray datasets. Through cross-correlation analysis, we estimate time-delayed interactions in *in silico*, *E. coli*, and *S. cerevisiae* networks, and show that SWING performs better than baseline methods in modules with more frequent time-delayed edges, such as the *tdcABC* regulon.

Interestingly, the apparent time delay only partially explains improved performance, as SWING also promotes edges without apparent time delays in *in silico* and *in vitro* networks. This discrepancy may arise from our conservative approach for identifying time delays; a more liberal approach could assign time delays to a greater fraction of the promoted edges. However, it is particularly challenging to estimate time delays for genes with multiple regulators using cross-correlation. More complex algorithms that incorporate additional information (*i.e.*, nonlinearity and partial correlation) could improve time delay estimation between regulators and targets[102].

An additional consideration involves interactions that occur faster than the sampling interval. These interactions will not exhibit a delay in the time series, and will resist inference and estimation of time delay regardless of methodology. This bottleneck can be managed by designing experiments with shorter sampling intervals. The choice of

sampling interval is context specific, and we recommend sampling with sufficient frequency to capture dynamics of interest.

2.4.2. SWING outperforms common network inference algorithms across scales

SWING outperforms common network inference algorithms——RF, LASSO, and PLSR——but is limited by computational expense. Since SWING constructs a larger explanatory matrix and executes multivariate comparisons between multiple time delays, it is more expensive than the aforementioned methods. Fortunately, SWING is trivially parallelizable and can be implemented on any multicore processing system. We conducted similarly derived 100-node *in silico* networks and found that SWING increased the AUPR and AUROC for all three methods (SI Appendix, Fig 2.S1), including SWING-LASSO, which had no significant difference for the 10-node networks (Fig. 2.2A). Remarkably, every single network was inferred with greater accuracy, indicating that SWING has notable benefits for larger inference tasks (SI Appendix, Fig 2.S1, Table 2.S1).

2.4.3. SWING is an extensible framework

Compared to other time-delayed inference algorithms, SWING is a flexible and extensible framework that is not limited to using a single statistical method. The SWING framework was implemented with RF, LASSO, and PLSR; it can be easily expanded to use other multivariate inference algorithms, including those that utilize prior information and heterogeneous data types[103]. Additional improvements can be made by incorporating complex weighting of methods for consensus analysis that leverage known weaknesses and biases of inference methods. Methods that involve empirical optimization of combination

weights, such as those assessed in the DREAM challenge, are expected to substantially improve SWING performance[104].

Although we implemented SWING to infer interactions from gene expression data, the same Granger causality principles can be applied to a wide variety of contexts with temporal dynamics. Provided sufficient time-series data, we expect SWING to identify regulatory relationships in related intracellular signaling pathways, as well as broader fields such as ecology, social sciences, and economics. As the sensitivity/specificity of experimental tools increases and the cost of implementation decreases, we expect longer and higher resolution time-series data to become widely available. We expect this increase in time resolution to further improve the accuracy of SWING-based network inference, especially as the community continues to build on the SWING chassis. The SWING framework, with currently implemented methods, is available on GitHub (*https://github.com/bagherilab/SWING*).

2.5. Supplementary Information

2.5.1. Materials and Methods

2.5.1.1. Gene regulatory network. SWING addresses the challenge of inferring regulatory networks from gene expression data. Gene regulatory networks are directed graphs with N nodes, where each node represents a gene. An edge from gene g_i to gene g_j indicates that g_i regulates the expression of g_j.

2.5.1.2. Time-series data. The time-series measurement of expression for gene, i, with T time points, is defined as $G_i = [g_i^1, g_i^2, \ldots, g_i^T]^\top$. Thus, a time-series experiment is defined as $\boldsymbol{T} = [G_1, \ldots, G_N]$. $\boldsymbol{T}$ is a $T \times N$ matrix which provides an ordered sequence

of values for each observed gene (columns) at each time point (rows).

$$\boldsymbol{T} = \begin{bmatrix} g_1^1 & \cdots & g_N^1 \\ \vdots & \ddots & \vdots \\ g_1^T & \cdots & g_N^T \end{bmatrix} \tag{2.1}$$

For simplicity we describe the case where there are no replicates. However, if there are multiple time series, P, of the same length for each gene, such as experiments with multiple biological replicates or experimental perturbations, they are stacked into a $(T \cdot P) \times N$ matrix such that $\boldsymbol{T} = [\boldsymbol{T}_1, ..., \boldsymbol{T}_p]^\top$.

2.5.1.3. SWING window creation. SWING employs a fixed-length sliding window to divide time-series observations into ensembles of training data with the same measured features within each time series.

Given a time-series data set $\boldsymbol{T}$, SWING creates Q consecutive windows. Q is defined as

$$Q = (T - w + 1)/s, \tag{2.2}$$

where w is the window width, such that $w \leq T$, and s is the step size between windows. Both w and s are specified by the user. Each window $\boldsymbol{W}_q$, where $q \in \{1, \ldots, Q\}$, is a subset of rows from the time-series data $\boldsymbol{T}$, such that:

$$\boldsymbol{W_q} = \begin{bmatrix} g_1^{s(q-1)+1} & \cdots & g_N^{s(q-1)+1} \\ g_1^{s(q-1)+2} & \cdots & g_N^{s(q-1)+2} \\ \vdots & \ddots & \vdots \\ g_1^{s(q-1)+w} & \cdots & g_N^{s(q-1)+w} \end{bmatrix} \tag{2.3}$$

If $w = T$ then there is only one window and SWING performs network inference equivalent to the base method. Additional parameters for window creation are described in the *SWING parameter selection*.

2.5.1.4. Edge inference. Once the temporal windows are delimited, we apply multivariate Granger causality to generate training sets for inference algorithms. Traditional Granger causality models assess pairwise predictions with a set delay between the variables. Previous methods expanded the Granger models to be multivariate, but do not simultaneously compare multiple delays between explanatory and response variables. Here we describe the formulation of a Granger model that is both multivariate and includes multiple delays.

SWING utilizes a general statistical framework where weights between explanatory variables and a response variable are calculated using supervised learning algorithms. For each window, W_q, we sequentially define a response vector for each gene, j, as $\boldsymbol{y}_j = \boldsymbol{W}_{q,j}$, which is the jth column of window $\boldsymbol{W}_q$. The explanatory data is created based on two user-specified parameters. The maximum lag, k_{max}, and minimum lag, k_{min}, define the number of time points that can exist between the explanatory variables and the response. They are used to define the user-allowed set of delays, $L = \{k_{min}, k_{min} + 1, \ldots, k_{max}\}$.

$|L|$ is the cardinality of the set L, and is used to calculate the maximum number of explanatory variables. For most windows the number of user-allowed delays is $|L| = k_{max} - k_{min} + 1$, but there will be fewer when $q \leq k_{max}$. The explanatory data matrix for each response vector is constructed by concatenating data from the delayed windows,

and is defined as

$$\boldsymbol{X} = \begin{cases} [\boldsymbol{W}_{q-k_{min}}, \ldots, \boldsymbol{W}_{q-k_{max}}] & q > k_{max} \\ [\boldsymbol{W}_{q-k_{min}}, \ldots, \boldsymbol{W}_1] & k_{min} < q \leq k_{max} \end{cases} \tag{2.4}$$

To maintain consistency between SWING and existing methods, if $k_{min} = 0$, the response variable is excluded from the explanatory data, prohibiting self-edges within the same window. $\boldsymbol{X}$ has an augmented number of explanatory variables, corresponding to an explanatory variable for each gene at each delay. The number of columns in $\boldsymbol{X}$ is $N \cdot |L|$ if $k_{min} > 0$, or $N \cdot |L| - 1$ if $k_{min} = 0$. We did not include any self edges, regardless of delay, during our testing, because the *in silico* and *in vitro* data was collected in a way that does not account for self-edges.

2.5.1.5. Model aggregation. SWING aggregates the results from several weak, but sensitive, windowed models to generate a ranked list of edges. Each window generates an $N \times (N \cdot |L|)$ adjacency matrix, $\boldsymbol{A}$, of edge scores where $A^k_{i,j}$ is the inferred score for gene i as the upstream regulator of gene j with delay k.

The time-series data are naturally left censored, as we cannot know measurements before the experiment occurs. As such, depending on the user specified k_{min} and k_{max}, some windows, particularly the earlier ones, will not infer interactions for larger values of k (*e.g.* $g_i^{q-2} \rightarrow g_j^q$ cannot be inferred if $q < 2$). Therefore, each window $\boldsymbol{W}_q$ infers at most $|L|$ scores, for each gene pair.

In order to combine scores across multiple windows and different delays into a single score $g_i \rightarrow g_j$, SWING does two aggregations. Confidence values from windowed subsets are aggregated into a single network by taking the mean rank of the edge at each delay k, and then taking the mean rank of the edge across all delays. Additionally, community

networks estimated from multiple classifiers are built by computing the mean rank of edges outputted from RF, LASSO, and PLSR. We use the edge rank because scores between window models and methods may not have equivalent distributions. The median of edge ranks may also be used, but in preliminary testing it did not significantly change the results.

2.5.1.6. SWING graph generation. A directed SWING graph shows causal relationships between N nodes in a system and can be represented by the adjacency matrix $\boldsymbol{A}$ in which each element $A_{i,j}$ is the confidence that an edge exists between parent node g_i and child node g_j. Given Q user-defined windows, for each window, W_q, there are at most $N^2|L| - N$ possible edges that exist in the inferred model. Therefore, the adjacency matrix for each window is

$$\boldsymbol{A_q} = \begin{bmatrix} A_{1,1}^{k_{min}} & \dots & A_{1,N}^{k_{max}} \\ \vdots & \ddots & \vdots \\ A_{N,1}^{k_{min}} & \dots & A_{N,N}^{k_{max}} \end{bmatrix}, \tag{2.5}$$

where $A_{i,j}^k$ is the confidence of the interaction whereby the parent node g_i is said to be Granger causal of the child node g_j with a delay of k time points. Self edges within the same window are prohibited, and therefore values $A_{i,i}^0$ are set to 0. In this way, a network model with N targets and at most $N \cdot |L|$ regulators is created for each window.

For each window, SWING estimates the confidence of each edge and generates a ranked list of edges based on method-specific criteria. Specifically, RF uses the importance score calculated with the mean squared error[105]; LASSO uses a stability selection metric[54], and PLSR uses the variable importance in projection (VIP) score[51]. The rank of an edge in each windowed model can be used as the confidence metric to compare across

methods. We compute a consensus model (SWING-Community) by calculating the mean rank across methods for each possible edge:

$$\overline{A}_{i,j} = \frac{R_{i,j}^{SWING-RF} + R_{i,j}^{SWING-LASSO} + R_{i,j}^{SWING-PLSR}}{3}, \tag{2.6}$$

where $R_{i,j}$ are the ranks of the edge for each of the tested methods, and $\overline{A}_{i,j}$ is the average rank of the edge $g_i \rightarrow g_j$ used as the confidence metric in the consensus network.

2.5.1.7. SWING parameter selection. SWING is a generalized framework that can be used with any multivariate machine learning inference method. In developing and testing SWING, we implemented three different existing methods: RF, LASSO, and PLSR. Each algorithm requires different tuning parameters. When using RF, we selected the number of trees, the maximum depth of the tree, and the number of trees based on guidelines from the GENIE3 manuscript[105]. For LASSO, we utilized two methods to select the regularization parameter[54]: for *in silico* studies, we selected the regularization parameter based on the cross-validation score; for *in vitro* data sets with comparatively less data, we selected the regularization parameter based on sensitivity analysis for a single random subnetwork and evaluated all subnetworks with the subsequent parameter. For PLSR, we selected the number of principal components to use based on the elbow criterion[51].

In addition to the base method's specific parameters, SWING has user-selected parameters that require knowledge of the system and data. For optimal performance, we suggest the window size be selected such that $T/2 \leq w \leq T$, where T is the number of time points in the time series. If $w < T/2$, increased noise can lead to inference of more false-positive edges. In general, the step size can be set to $s = 1$, unless the user has an abundance of time points and wishes to train on only a subset of the data.

The *in silico* data from GNW is generated such that the perturbation is applied before the simulation and removed at $T/2$. We therefore used $w = 0.5T \approx 10$, to capture the change in dynamics based on the perturbation. For consistency, we also used $w = 0.5T = 7$ for the *in vitro E. coli* SOS network inference.

The allowed delay range is specified by the user in setting k_{max} and k_{min}. We recommend the user set these values based on the range of dynamics expected in the system, or by prior delay analysis such as cross-correlation. Since k_{max} and k_{min} are integer values, they also depend on the sampling interval of the experimental data. Specifying $k_{min} = 0$ allows SWING to infer edges with no delay, as many existing methods do. When testing the *in silico* networks we used $k_{max} = 3$ and $k_{min} = 1$, corresponding to an allowed delay range of $50 \leq k \leq 150$ minutes based upon the *in silico* sampling strategy. This range is consistent with the delays in the *in silico* data estimated using cross-correlation. If, however, the user specifies null SWING parameters—specifically, $w = T$, $k_{max} = 0$, $k_{min} = 0$, and $s = 1$—there is only a single window with no delays between the explanatory and response variables. This condition corresponds to running the base methods independent of SWING.

2.5.1.8. *In silico* data generation. All *in silico* networks were created using GeneNetWeaver[61]. GNW creates a stochastic differential equation (SDE) model from which time-series data are sampled. The kinetic models incorporate Hill kinetics and include both transcriptional and translational components. We generated time-series perturbations for 20 non-isomorphic, 10-node and 100-node subnetworks from the curated *E. coli* and *S. cerevisiae* networks. Simulated data includes ten random combinations of perturbations which are uniformly sampled at 21 time points with a maximum time of 1000 in arbitrary units.

2.5.1.9. Parameters for GNW subnetwork extraction. GeneNetWeaver (GNW) is designed to provide synthetic benchmarking data sets for the assessment of network inference methods. GNW includes the networks used for assessment in the DREAM4 challenge, as well as *E. coli*-derived and *S. cerevisiae*-derived gene regulatory networks, which can be used to extract testable subnetworks[106]. These features make GNW ideal for generating *in silico* gene expression data paired with an unambiguous gold standard.

We extracted subnetworks from curated *E. coli*-derived and *S. cerevisiae*-derived networks included in GNW. For each model organism we extracted 20 non-isomorphic networks with 10 and 100 nodes. All subnetworks were extracted with neighbors chosen via greedy selection. The *S. cerevisiae*-derived subnetworks were extracted with 50% of the nodes chosen from the strongly connected component. The curated *E. coli*-derived network does not have one strongly connected component, and therefore *E. coli*-derived subnetworks were extracted starting with a randomly selected vertex. To ensure uniqueness of subnetworks, each sequential network is randomly extracted and preserved only if it is non-isomorphic to all previously extracted networks.

Time-series perturbation data was generated for each of the extracted subnetworks using the default DREAM4 challenge parameters included in GNW. Simulated data includes ten random combinations of perturbations. Simulated experimental perturbations are applied immediately before the time-series data is sampled, and removed halfway through the simulation.

2.5.1.10. *In silico* predictions and scoring. We scored the inferred networks by calculating the mean increase in the area under the precision-recall (AUPR) curve and the area under the receiver operator characteristic (AUROC) curve for N networks as

follows:

$$\overline{S}_{increase} = \frac{\sum_{n=1}^{N} S_{n,SWING} - S_{n,base}}{N}, \tag{2.7}$$

where S_n is the AUPR or AUROC for an individual network, n. For a stochastic method, like RF, S_n, is the mean AUPR/AUROC over several trials, for an individual network.

2.5.1.11. Cross-correlation and lag analysis. Temporal cross-correlation has been used by multiple studies to describe how well two signals are correlated when one is shifted in time relative to the other[95 90]. Let $G_i = [g_i^1, g_i^2, \ldots, g_i^T]$ represent measurements of a single gene in a time-series data set. We calculated the pairwise cross-correlation, R, between a pair of signals, G_i and G_j, for a delay k as:

$$R^k_{G_i,G_j}(t) = \frac{\sigma_{G_iG_j}(t)}{\sigma_{G_i}\sigma_{G_j}} \tag{2.8}$$

σ_{G_i}, σ_{G_j}, and $\sigma_{G_iG_j}(t)$ refer to the standard deviation of G_i, standard deviation of G_j, and cross-covariance of G_i and G_j at time t, respectively. The cross-covariance is defined by:

$$\sigma_{G_iG_j}(t) = \frac{1}{N-1}\sum_{t=1}^{N}(G_{i,t-k} - \mu_{G_i})(G_{j,t} - \mu_{G_j}), \tag{2.9}$$

where μ_{G_i} and μ_{G_j} define the mean values of each time series.

We applied several stringent criteria to evaluate time-delayed edges. We calculated the two-sided p-value using the t-distribution equation and subsequently corrected the p-value using the Bonferroni correction (the significant p-values were those less than $\alpha \leq \frac{0.05}{m}$ where m is the total number of edges evaluated)[107]. Since multiple experiments were evaluated for each pairwise comparison, we filtered noisy lagged edges by removing edges in which the sign of the lag differed in more than 10% of experimental perturbations. For

E. coli and *S. cerevisiae in vitro* data, we also incorporated prior knowledge regarding the sign of the interaction into the lag selection. If multiple delays were significant, depending on whether the parent positively or negatively regulated the target in the gold standard, we selected the lag with the smallest p-value that maximized ($0 < R < 1$) or minimized ($-1 < R < 0$) cross-correlation, respectively. We evaluated cross-correlation at $k = \{0, 10, 20, 30, 60, 90\}$ in *E. coli* and *S. cerevisiae* data sets.

2.5.1.12. *In vitro* data aggregation. We extracted *in vitro* gold standard networks for *E. coli* and *S. cerevisiae* from RegulonDb and DREAM5 Yeast gold standards (Network4) respectively[47]. For *E. coli*, we extracted the known set of TF and gene interactions from RegulonDb 9.0[108]. To derive subnetworks from parent gold standards, we performed MCODE clustering using modularity parameters of 0.25 (*E. coli*) and 0.5 (*S. cerevisiae*), resulting in subnetworks where the number of nodes in each module is between 3 and 145 (Tables 2.S4 and 2.S7). Gene ontology enrichment analysis was performed using a cutoff for false discovery rate-corrected $p < 0.05$ and the *goatools* package[109].

Sources of time-series data sets for *E. coli* and *S. cerevisiae* are described in Tables 2.S2 and 2.S6. To run SWING, 10 minute time points were generated using cubic spline interpolation and this data was used to train both SWING and baseline methods[110]. Data interpolation was not needed for lag analysis in Figs. 2.4A and 2.S8. Time-series data sets were mean centered.

2.5.1.13. Computational development. The SWING package was developed in Python 3.4.5 using the following major packages: *NumPy and SciPy*[111], *pandas*[112], and *NetworkX*[113]. The RF, LASSO, and PLSR algorithms use implementations available in *scikit-learn*[114]. Figures were generated using *seaborn* and *matplotlib*[115]. The code for SWING can be found on GitHub (*https://github.com/bagherilab/SWING*).

2.5.2. Sensitivity Analysis

We conducted sensitivity analysis to assess SWING performance (AUPR and AUROC) as a function of user-specified parameters: window size (w), minimum lag (k_{min}), and maximum lag (k_{max}). We also assessed performance as a function of parameters relating to experimental design: sampling interval of the time series, as well as mixed interval time-series samples. In our analysis, we systematically compared frontline SWING methods to baseline methods (SWING-RF to RF, SWING-LASSO to LASSO, SWING-PLSR to PLSR) to determine the range of SWING performance for 40 10-node *in silico* networks, and the *E. coli* SOS network data set analyzed in Figure 2.3. We quantified this performance for every possible window size that can be specified, and a wide range of k_{min}/k_{max} combinations. The percent change in AUPR or AUROC for an individual network is calculated as:

$$S_{\%change} = 100 * \frac{S_{SWING} - S_{base}}{S_{base}}, \tag{2.10}$$

where S is the AUPR or AUROC. For a stochastic method, like RF, S_{base} and S_{SWING} are the mean scores over several realizations. We also describe experimental design considerations related to time-series sampling interval.

2.5.2.1. Effect of varying window size. Overall, SWING performance is higher than corresponding baseline methods for three frontline inference methods over a wide range of window sizes (Fig. 2.S9), however, the performance of a single network (red line) may differ between individual networks and SWING methods. For SWING-RF, the mean and median percent change of AUPR and AUROC for 40 *in silico* networks is greater than RF for all possible window sizes, $w : w \in \{2, \ldots, 20\}$. The distribution range indicates that

the performance for any individual network can be lower than baseline, and may partly depend on unknown properties, such as underlying network structure.

We highlight an individual network (Fig. 2.S9, red line) for which SWING-RF AUPR is slightly higher than RF, but SWING-RF AUROC is lower when $w = 2$. SWING-RF steadily outperforms RF with larger window sizes until it peaks at $w = 17$. The majority of the AUPR/AUROC distributions show SWING-RF outperforming the baseline, indicating that SWING improves inference of many network topologies. SWING-LASSO and SWING-PLSR improvements are more modest, and SWING-LASSO performance is less stable across window sizes. Nevertheless, SWING generally outperforms the baseline method, and rarely performs worse.

Similarly, in the *in vitro E. coli* SOS network analyzed in the main text (Fig. 2.3), we demonstrate that choosing $w : w \in \{5, \ldots, 12\}$ increases AUPR compared to baseline ($w = 14$), and choosing $w : w \in \{5, \ldots, 13\}$ increases AUROC (Fig. 2.S13B). Thus, the optimal window size partially depends on the underlying inference method and the individual network structure, but SWING appears to improve overall network inference capability across a wide range of user-defined parameters.

2.5.2.2. Effect of varying minimum and maximum lag. SWING outperformed baseline methods for a wide range of k_{min}/k_{max} combinations (Fig. 2.S10), but again, the performance of a single network may differ depending on the inference method used or underlying network properties. The aggregate mean and median AUPR and AUROC for SWING methods was greater than that of their respective baselines for almost all k_{min}/k_{max} combinations (Fig. 2.S10). Generally, specifying too small of a k_{min}/k_{max} range may result in overlooking a majority of lagged interactions, while too large of a k_{min}/k_{max} range may result in spurious inference. Trends are distinct between inference methods:

SWING-LASSO appears to be sensitive to over-specifying k_{min}/k_{max} combinations, as combinations with a smaller range (*e.g.* $k_{min} = 1$ and $k_{max} = 1$) tend to outperform larger ranges, while SWING-RF and SWING-PLSR appear to be more robust to larger k_{min}/k_{max} ranges.

We performed cross-correlation lag analysis to show that the distribution of apparent delays in the underlying network partly determines the optimal lag parameters. The apparent time delay of the majority of interactions in the *in silico* networks (sampling interval of 50 minutes) is between 0 and 100 minutes which corresponds with k_{min}/k_{max} combination (0, 2). Less than 20% of interactions have apparent delays greater than 100 minutes (Fig. 2.S14). In SWING-PLSR and SWING-RF, optimal mean performance occurs for k_{min}/k_{max} combinations that tend to capture longer delays, while for SWING-LASSO, optimal performance occurs at k_{min}/k_{max} combinations that tend to capture shorter delays (Fig. 2.S10).

In the *E. coli* SOS network, cross-correlation lag analysis only identified a few edges with apparent lag, (Fig. 2.3) all with a delay of one time point. By varying k_{min}/k_{max}, we show that SWING-RF tends to perform well for combinations with k_{min} of 0 or 1, and k_{max} of 0 or 2 (Fig. 2.S13A). Combinations that overspecify a large range of possible lags ($k_{min} = 0$ and $k_{max} = 3$ for instance), or stray too far from the underlying lag distribution, perform poorly.

2.5.2.3. Effect of sampling interval. The sampling interval of the data set strongly affects whether SWING can be effectively utilized to improve network inference. We generated 40 finely sampled *in silico* networks and sampled time points such that resulting time-series data sets had intervals of 10 (finest), 30, 50, 100, 200, 333, 500 (coarsest) minutes. We also performed cross-correlation lag analysis on sampled data sets to show

the underlying apparent lag distribution captured by sampling strategies. To compare the performance of SWING between different sampling intervals, we utilized the following parameters: $w = 0.66T$, $k_{min} = 1$, $k_{max} = 3$.

Sampling intervals greater than 100 minutes show no apparent time delay (Fig. 2.S14). Concomitantly, SWING performance compared to baseline tends to be poor for sampling intervals greater than 100 minutes, though this observation is method-dependent (Fig. 2.S11A). While on average, SWING-LASSO and SWING-PLSR perform poorly with such long sampling intervals, individual networks do perform well (as indicated by the large variance). As expected, SWING performance is compromised when the k_{min}/k_{max} are incorrectly specified even with finer sampling. With a sampling interval of 10 minutes, a majority of time-delayed interactions show an apparent lag between 10 minutes and 80 minutes (Fig. 2.S14). SWING methods still tend to have a higher mean and median AUPR/AUROC than baseline methods with a k_{min} of 10 minutes and a k_{max} of 30 minutes, but performance was greatly improved when k_{min}/k_{max} were changed to include a majority of underlying time delays (Fig. 2.S11B). In summary, SWING outperforms baseline methods over a wide range of sampling intervals. However, as expected, when the sampling interval decreases, SWING performance also decreases (Fig. 2.S11A). We attribute this observation to a decrease of the number of true edges for which an apparent lag can be estimated (Fig. 2.S14).

We also tested the effect of uneven sampling on SWING performance. For an equal comparison, the same number of time points were selected for both strategies, spanning roughly the same time frame. In general, both uneven and even sampling strategies enabled SWING-RF, SWING-LASSO, and SWING-PLSR to outperform than their respective baseline methods (Fig. 2.S12). Additionally, in the event that the user wishes

to input data from a mixed design, data from different steady state time points (t=0 or some long time after perturbation) would need to be labeled as such when inputting the raw data points into the algorithm.

2.5.3. Supplementary Information: Additional Figures

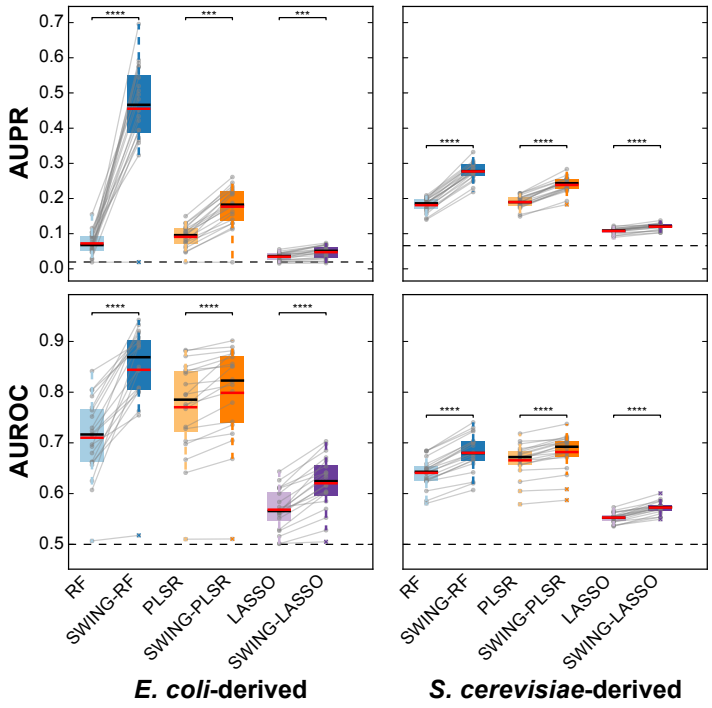


Figure 2.S1. Changes in AUPR and AUROC curve distributions for 100-node GNW networks. All networks and methods show a significant improvement in both the AUPR and AUROC when using SWING. Score changes to individual networks are shown as grey lines. The mean (red) and median (black) of each score distribution is shown. The expected score based on random for each metric is shown as a dashed line. n=20 networks, k_{min}=1, k_{max}=3, and w=10 for all networks. p-values are calculated using the Wilcoxon signed-rank test, **** p<0.0001, *** p<0.001.

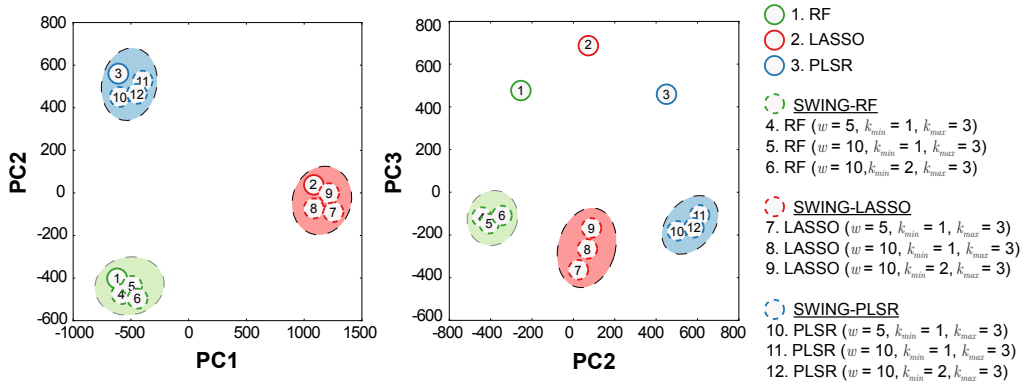


Figure 2.S2. SWING and non-SWING methods are grouped according to similarity of ranked predictions for 40 10-node *in silico* networks via principal component analysis. The first (58% variance explained) and second (15% of variance explained), and the second and third (5% of variance explained) principal components are shown. The first principal component largely separates inference methods based on performance, while the second component separates methods based on underlying base method. The second and third principal components seem to explain motif biases. Networks inferred by various SWING parameter selections cluster together according to inference type, with SWING methods forming clusters distinct from corresponding base methods.

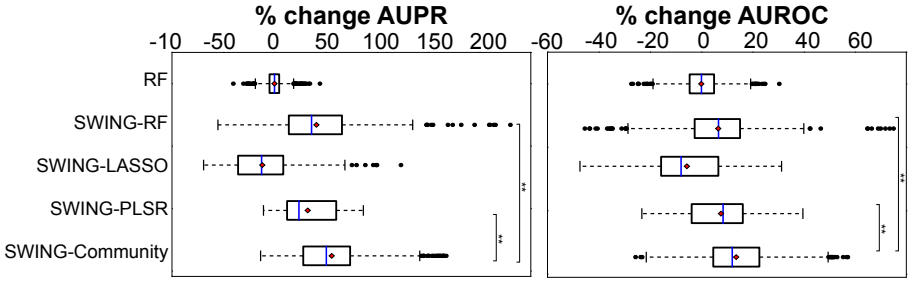


Figure 2.S3. Boxplots show the percent change in performance of RF vs SWING-RF, SWING-LASSO, SWING-PLSR, SWING-Community prediction for 40 10-node networks. The percent change of AUPR and AUROC from RF was calculated for each network (500 trials). For SWING methods, the following parameters were used: $w = 10$, $k_{min} = 1$, and $k_{max} = 3$. p-values are calculated using the Mann-Whitney U test, ** p <0.01.

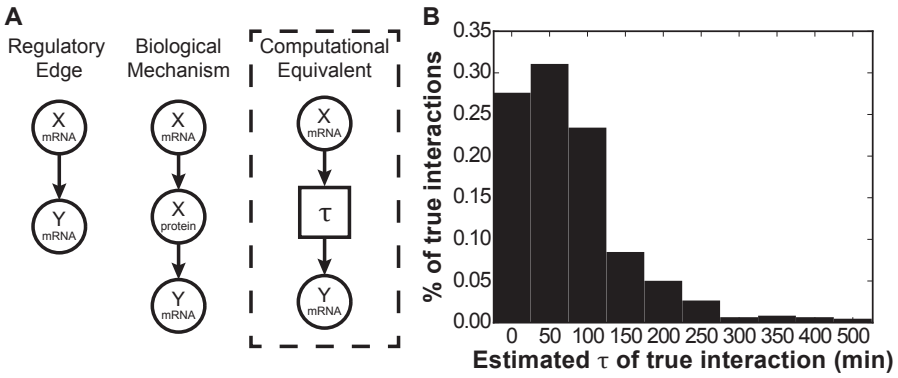


Figure 2.S4. Identification of delays in DREAM 4 *in silico* networks. (A) Real biological systems require additional steps, such as translation, whose kinetics determine the delay between upstream gene expression and downstream nodes. The physical time required for these steps can be expressed as a delay, τ, and must be accounted for when inferring GRNs from gene expression data. (B) The τ for each edge in five DREAM 4 *in silico* networks was calculated using the cross-correlation function. $\tau = 0$ minutes was calculated for around 25% of the interactions, indicating no delay could be identified. A large fraction of interactions have $50 \leq \tau < 150$ min, indicating that the kinetics of the model result in a delay. Larger values of τ may be due to the kinetics, but very high values are likely due to noise.

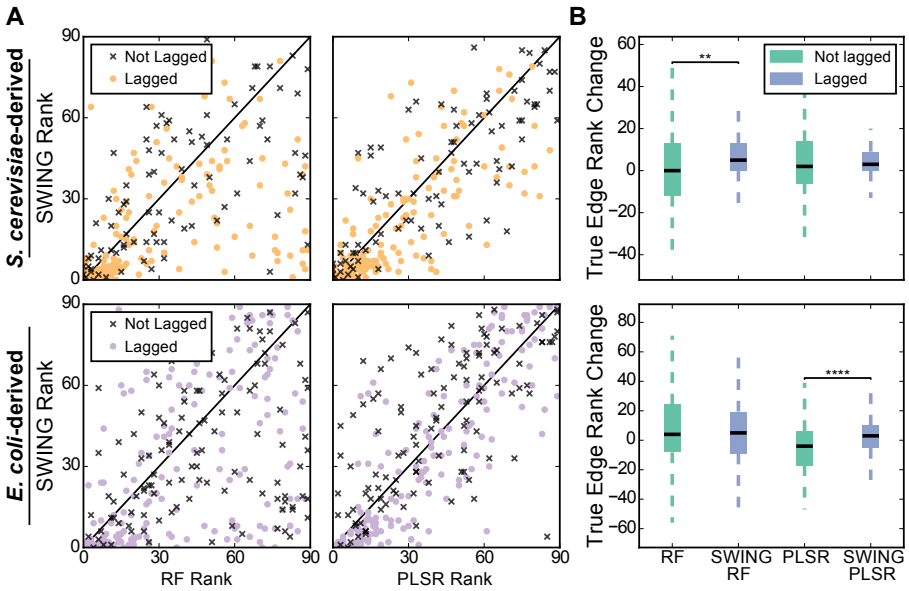


Figure 2.S5. SWING promotes edges with apparent time delays between genes. (A) The inferred rank using SWING versus the respective control methods. All true edges for the 20 networks are plotted. Edges that present below the diagonal line are promoted by SWING, and those that present above are demoted. For all methods and networks, SWING is significantly more likely to promote an edge with apparent lag. (B) Distribution of true edge rank changes when using SWING for lagged and not lagged edges. The median true edge promotion of lagged edges is significantly greater for *E. coli* networks when SWING is run using PLSR, but significantly greater for *S. cerevisiae* networks when SWING is run using RF. p-values are calculated using the Mann-Whitney U test, **** $p < 0.0001$, ** $p < 0.01$. n=292 for *E. coli* and n=257 for the *S. cerevisiae*.

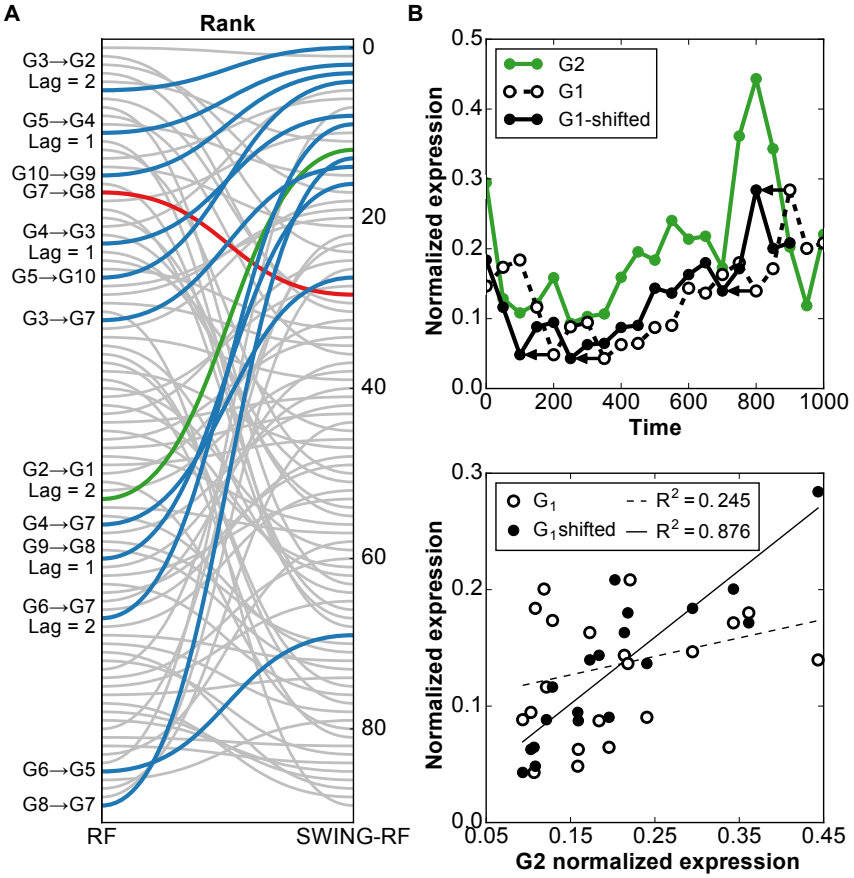


Figure 2.S6. SWING promotes time-delayed edges and increases correlation between genes. (A) Edge rank comparison for *S. cerevisiae* network 12 when using SWING-RF compared to RF (blue = promoted edges, red = demoted edges, grey = false edges, green = $G2 \rightarrow G1$ analyzed in panel B). Edges for which a time delay could be estimated are labeled. (B) Improved correlation between G2 and G1 when a lag is artificially introduced.

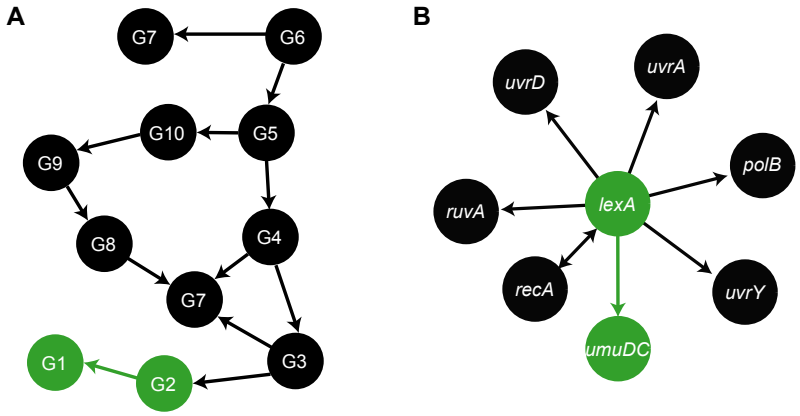
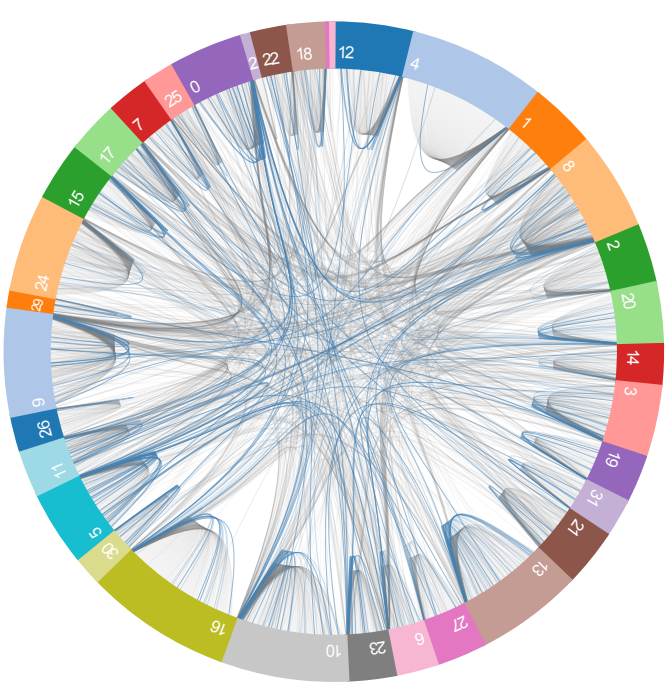


Figure 2.S7. Graph representations of (A) *S. cerevisiae*-derived network 12 and (B) *E. coli* SOS network. Nodes and edges highlighted in green are specifically interrogated in Figs. 2.S6 and 2.3.



ID Ontology
0 glycogen metabolic process
1 unassigned
2 oxidation-reduction process
3 transcription, DNA-templated
4 D-gluconate metabolic process
5 carbohydrate transport
6 iron ion homeostasis
7 sulfate assimilation
8 oxidation-reduction process
9 lipid metabolic process
10 cellular amino acid biosynthetic process
11 membrane
12 cellular response to DNA damage stimulus
13 unassigned
14 phosphate ion transport
15 unassigned
16 glyoxylate catabolic process
17 response to arsenic-containing substance
18 unassigned
19 biotin biosynthetic process
20 cyanate catabolic process
21 glycerol metabolic process
22 unassigned
23 aromatic compound catabolic process
24 unassigned
25 potassium-transporting ATPase activity
26 unassigned
27 NAD biosynthetic process
28 phage shock
29 unassigned
30 thiamine transport
31 cellular amino acid biosynthetic process
32 nucleobase-containing small molecule metabolic process
33 unassigned
34 unassigned
35 zinc II ion transport

Figure 2.S8. Cross-correlation analysis of time-delayed interactions derived in *S. cerevisiae*. Circular diagram depicts experimentally validated interactions and gene ontologies present in each module. Blue edges depict edges displaying time-delayed interactions (time delay of 10 minutes or greater) inferred using pairwise cross-correlation from curated microarray data.

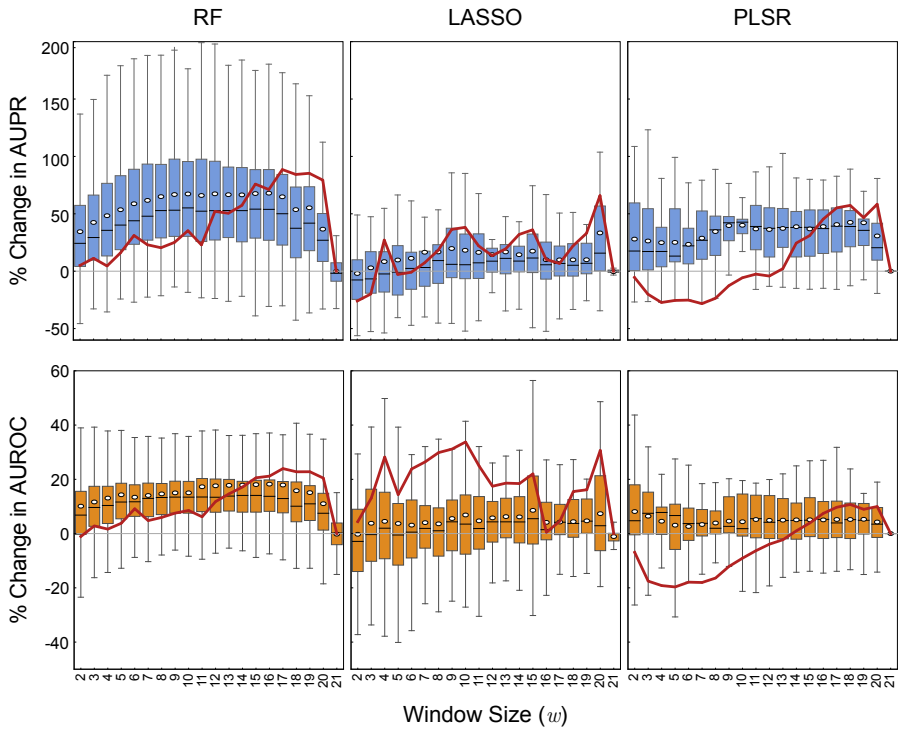


Figure 2.S9. Boxplots show the percent change in performance of SWING-RF, SWING-LASSO, SWING-PLSR compared to baseline methods (RF, LASSO, PLSR respectively) for each window size change (white dot = mean, black bar = median). The percent change of AUPR and AUROC from non-SWING methods was calculated for each network (100 trials, 40 10-node networks). The red line indicates the AUROC/AUPR for one example network as w changes. For SWING methods, the following parameters were used: $k_{min} = 1$, and $k_{max} = 3$ (k_{max} was adjusted accordingly to be the largest allowed value when w was 19, 20, and 21).

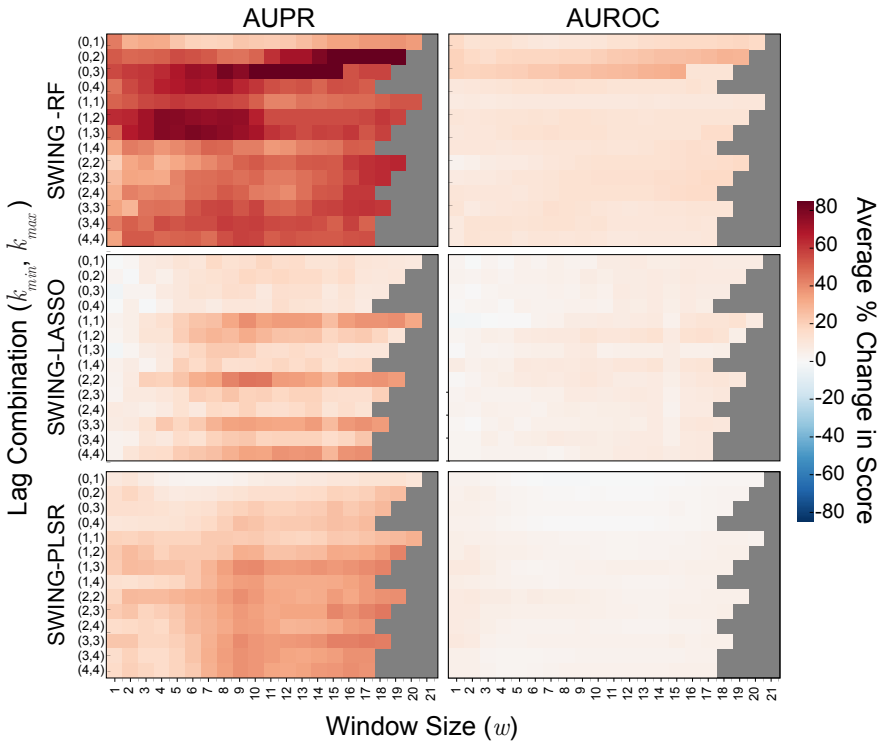


Figure 2.S10. Results of min/max sensitivity analysis on 40 *in silico* networks using SWING-RF, SWING-LASSO, and SWING-PLSR. Heatmaps show the percent change of a three parameter scan (w, k_{min}, and k_{max}) compared to the respective baseline. The mean performance change was determined for 50 realizations for each network, then the mean of performance change for 40 10-node networks was plotted. Red denotes that SWING performed better than the corresponding baseline method. Parameter combinations that are not possible are shown in grey.

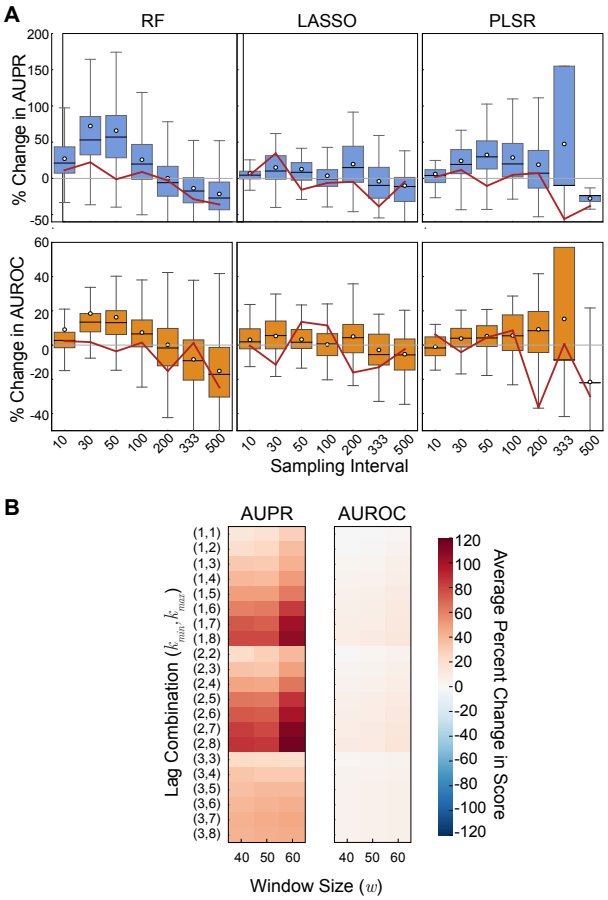


Figure 2.S11. Results of sampling interval analysis on 40 *in silico* networks using SWING and baseline methods. (A) Boxplots show the percent change in performance of SWING-RF, SWING-LASSO, SWING-PLSR compared to baseline methods (RF, LASSO, PLSR respectively) for each sampling interval (white dot = mean, black bar = median). For example, the sampling interval 333 indicates that samples were taken at t=0, 333, 666, and 999. The percent change of AUPR and AUROC from non-SWING methods was calculated for each network (100 trials). The red line indicates the AUPR/AUROC for one example network as w changes. For SWING methods, the following parameters were used: $k_{min} = 1$, and $k_{max} = 3$ (k_{max} was adjusted accordingly to be the largest allowed value when w was 19, 20, and 21), the window size was selected to be an integer $\frac{2}{3}$ of the total window size rounded down. (B) Corresponding heatmap for parameter scan where data was subsampled at 10 minute intervals for SWING-RF relative to RF. This data demonstrates that using k_{min} and k_{max} values that are inclusive of underlying lag distribution substantially increases performance.

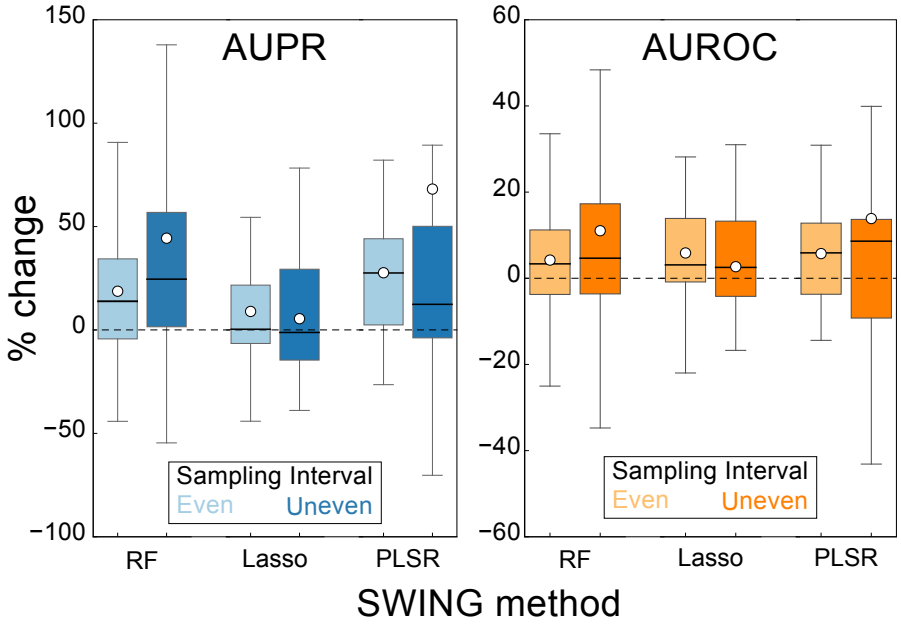


Figure 2.S12. Boxplots show the percent change in performance of SWING-RF, SWING-LASSO, SWING-PLSR compared to baseline methods (RF, LASSO, PLSR respectively) for even vs. uneven sampling intervals (white dot = mean, black bar = median) of 40 10-node *in silico* networks. For the uneven sampling strategy, we selected 8 time points t = 0, 15, 30, 60, 120, 240, 480, 720. For the even sampling strategy, we selected 8 time points t = 0, 100, 200, 300, 400, 500, 600, 700.The percent change of AUPR and AUROC from non-SWING methods was calculated for each network (100 trials). For SWING methods, the following parameters were used: $k_{min} = 1$, and $k_{max} = 3$, $w = 4$.

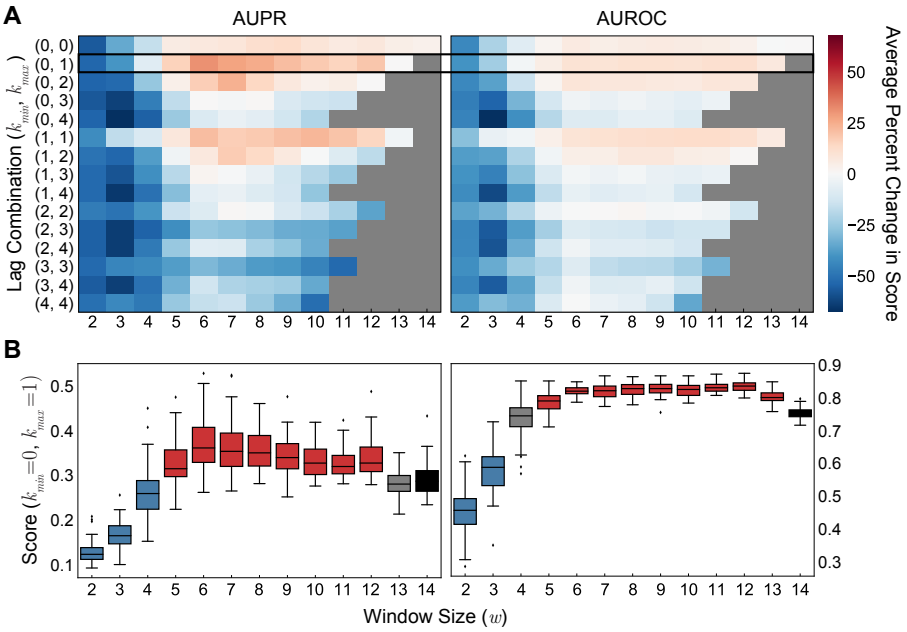


Figure 2.S13. Results of sensitivity analysis on *in vitro* SOS data using SWING-RF. (A) Heatmaps show performance change of a three parameter scan (w, k_{min}, and k_{max}) of the *in vitro* SOS network. Red denotes that SWING-RF performed better than the RF baseline method with the indicated parameters while blue denotes that SWING-RF performed worse than the RF baseline method. Parameter combinations that are not possible are shown in grey. (B) Boxplots show AUPR and AUROC distributions for 50 trials of SWING-RF at each window size, w, with $k_{min} = 0$, $k_{min} = 1$. These plots show an example of the variance AUPR/AUROC scores for each RF realization, for the row outlined in A (black = baseline distribution using RF; blue = distributions with a significantly lower score than the baseline; red = distributions with a significantly higher score than the baseline; grey = distributions with no significant score difference than the baseline. p-values are calculated with a paired t-test. Values are considered significant with $p < 0.05$).

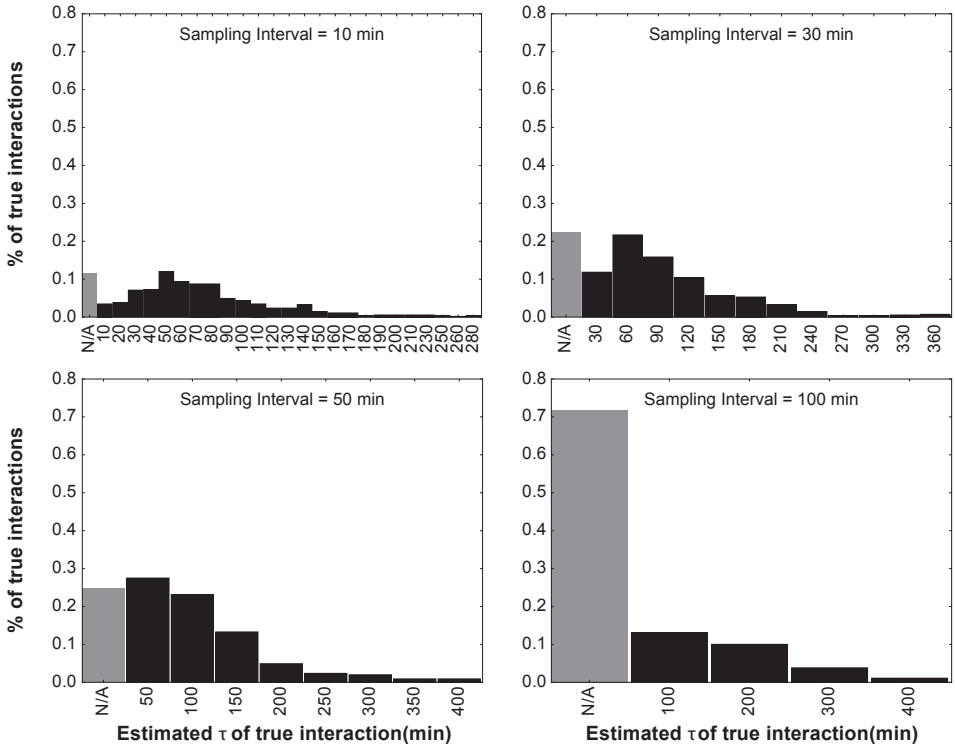


Figure 2.S14. Barplots show the lag distribution each sampling interval for each aggregated *in silico* data sets. For example, the sampling interval 333 indicates that samples were taken at t=0, 333, 666, and 999. Lag was calculated using cross-correlation for 40 10-node networks, 20 *E coli*-derived and 20 *S. cerevisiae*-derived. True edges for which no apparent lag was calculated are labeled as "N/A".

		AUPR			AUROC		
Dataset	**Method**	Δ	Δ (%)	*p*-value	Δ	Δ (%)	*p*-value
Ecoli10	SWING-RF	0.151	98.1	1.20E-04	0.120	25.5	2.93E-04
Ecoli10	SWING-PLSR	0.072	35.7	3.90E-04	0.028	5.4	1.35E-01
Ecoli10	SWING-LASSO	0.023	13.1	1.37E-02	0.037	7.7	6.42E-03
Scerevisiae10	SWING-RF	0.139	50.4	1.20E-04	0.058	8.3	6.81E-04
Scerevisiae10	SWING-PLSR	0.102	35.1	1.40E-04	0.035	5.1	1.37E-02
Scerevisiae10	SWING-LASSO	0.001	0.6	6.54E-01	-0.002	0.0	7.94E-01
Ecoli100	SWING-RF	0.383	647.9	8.86E-05	0.134	19.1	8.86E-05
Ecoli100	SWING-PLSR	0.086	96.2	1.03E-04	0.028	3.7	8.86E-05
Ecoli100	SWING-LASSO	0.013	37.3	8.86E-05	0.052	9.1	8.86E-05
Scerevisiae100	SWING-RF	0.096	53.3	8.86E-05	0.040	6.1	8.86E-05
Scerevisiae100	SWING-PLSR	0.049	25.7	8.86E-05	0.016	2.4	8.86E-05
Scerevisiae100	SWING-LASSO	0.012	11.6	8.86E-05	0.020	3.6	8.86E-05

Table 2.S1. Summary of SWING performance on *in silico* networks. The change in mean AUPR and AUROC for 20 *in silico* 10 and 100-node networks. *p*-values are calculated using the Wilcoxon signed-rank test.

Strain/Condition	Time Points	Citation
MG1655 control	t=10,20,30,40,50	[116]
MG1655 cold stress	t=10,20,30,40,50	[116]
MG1655 heat stress	t=10,20,30,40,50	[116]
MG1655 oxidative stress	t=10,20,30,40,50	[116]
EMG2 LB 0pt02percent glucose	t=150,180,210,240,270,300,330,360,480	[47]
EMG2 LB 0pt04percent glucose	t=150,180,210,240,270,300,330,360,480	[47]
MG1655 wt untreated	t=0,30,60,90,120	[47]
MG1655 wt MMC 2pt5ug	t=0,30,60,90,120	[47]
MG1655 wt UV 500en	t=0,30,60,90,120	[47]
BW25113 uninduced	t=0,30,60,120,180	[47]
BW25113 norflaxacin	t=0,30,60,120,180	[47]
BW25113 D recA	t=0,30,60,120,180	[47]
BW25113 U ccdB	t=0,30,60,120,180	[47]
BW25113 D recA norflaxacin	t=0,30,60,120,180	[47]
BW25113 D recA U ccdB	t=0,30,60,120,180	[47]
MG1655 U lacZ	t=0,30,60,90	[47]
MG1655 U ccdB	t=0,30,60,90	[47]
EMG2 LB norf 25ng	t=0,12,24,36,48,60	[117]

Table 2.S2. *E. coli* data set for RegulonDB lag analysis

Cluster ID	Total # Edges	# of Lagged Edges ($k \geq 10m$)	% Lagged Edges
0	125	8	6%
1	542	138	25%
2	193	14	7%
3	113	24	21%
4	13	4	31%
5	299	87	29%
6	55	5	9%
7	132	11	8%
8	65	8	12%
9	71	2	3%
10	124	29	23%
11	106	8	8%
12	76	6	8%
13	203	17	8%
14	36	3	8%
15	56	2	4%
16	18	8	44%
17	102	8	8%
18	141	15	11%
19	36	3	8%
20	30	1	3%
21	32	7	22%
22	86	10	12%
23	27	12	44%
24	204	25	12%
25	167	20	12%
26	94	14	15%
27	45	3	7%
28	43	3	7%
29	34	2	6%
30	271	33	12%
31	114	19	17%
32	29	8	28%
33	29	1	3%
34	23	0	0%

Table 2.S3. Lagged edge analysis of 35 *E. coli* subnetworks from RegulonDB. We highlight lagged edges with apparent time delays of 10 minutes or greater.

Cluster ID	Gene Ontology	Significance (Corrected P-Value)	# of Genes in GO	Size of GO Category
1	putrescine catabolic process	4.86E-05	7	110
2	iron ion homeostasis	9.24E-25	20	74
3	metal ion binding	0.008352298	20	41
3	oxidation-reduction process	0.002072459	18	41
4	rhamnose metabolic process	1.72E-10	5	6
5	ATP-binding cassette (ABC) transporter complex	0.004500043	8	43
5	chemotaxis	4.42E-05	7	43
6	purine nucleotide biosynthetic process	5.01E-20	12	25
7	cellular response to DNA damage stimulus	2.28E-16	30	63
8	organic phosphonate catabolic process	2.20E-12	8	40
9	glycogen metabolic process	0.000432348	4	29
10	tryptophan catabolic process	0.027249226	3	37
11	cellular amino acid biosynthetic process	0.028322085	9	50
13	translation	2.29E-06	12	40
13	intracellular ribonucleoprotein complex	7.77E-08	11	40
14	DNA replication	0.000287344	6	16
15	arginine biosynthetic process	1.17E-18	11	33
16	propionate catabolic process, 2-methylcitrate cycle	3.84E-06	4	10
17	drug transmembrane transport	0.000254503	7	47
18	bacterial-type flagellum organization	1.70E-06	8	53
19	leucine biosynthetic process	1.16E-06	5	16
20	galactose metabolic process	1.20E-09	5	10
21	D-galacturonate catabolic process	0.009971194	3	13
22	carbohydrate transport	5.04E-12	16	37
23	L-threonine catabolic process to propionate	1.47E-12	6	8
25	oxidation-reduction process	1.07E-05	24	54
26	response to copper ion	1.32E-07	6	20
27	sulfate assimilation	8.24E-09	7	29
28	nucleoside transport	0.005095171	3	14
29	fatty acid metabolic process	1.94E-27	16	21
30	oxidation-reduction process	2.21E-11	32	59
32	D-gluconate metabolic process	2.80E-12	7	13
33	cellular amino acid biosynthetic process	1.90E-07	10	19
34	aromatic amino acid family biosynthetic process	8.37E-23	13	20

Table 2.S4. Gene ontological analysis of *E. coli* subnetworks in RegulonDB

Transcription Factor	Total # Edges	# of Lagged Edges ($k \geq 10m$)	% Lagged Edges
crp	466	129	28%
fnr	286	34	12%
csgd	24	7	29%
fis	166	6	4%
torr	11	6	55%
lexa	52	5	10%
iscr	27	5	19%
arca	155	4	3%
gntr	11	4	36%
narl	120	3	3%
soxs	32	3	9%
fliz	19	3	16%
oxyr	32	2	6%
cysb	28	2	7%
argp	13	2	15%
fur	117	1	1%
cpxr	55	1	2%
narp	50	1	2%
pdhr	35	1	3%
purr	29	1	3%
rcsb	20	1	5%
fadr	18	1	6%
evga	15	1	7%
mraz	15	1	7%
arac	13	1	8%
leuo	12	1	8%
basr	12	1	8%
lrp	69	0	0%
phob	54	0	0%
mode	46	0	0%
phop	39	0	0%
argr	35	0	0%
mara	31	0	0%
nagc	30	0	0%
fhla	29	0	0%
gade	26	0	0%
gadx	26	0	0%
rob	20	0	0%
nac	18	0	0%
metj	15	0	0%
ydeo	15	0	0%
ompr	14	0	0%
gadw	12	0	0%
paax	12	0	0%
cytr	12	0	0%
yedw	12	0	0%
trpr	11	0	0%
cusr	11	0	0%
hyfr	11	0	0%
tyrr	11	0	0%

Table 2.S5. Lagged edge analysis of *E. coli* transcription factors from RegulonDB. We highlight lagged edges with apparent time delays of 10 minutes or greater.

Strain/Condition	Time Points	Citation
Y262 Wild type cells, oxidative stress	t=0,30,60,100,140,180	[118]
Y262 Wild type cells, DNA damage stress	t=0,30,60,100,140,180	[118]
Y262 Wild type cells, oxidative decay	t=0,5,10,15,20,30,40,50,60	[118]
Y262 Wild type cells, DNA damage decay	t=0,5,10,15,20,30, 40, 50, 60	[118]
IFO0233 Wild type cells, control	t = 830,834,838,842,846,850,854,858,862,866,870	[119]
IFO0233 Wild type cells, phenelzine treatment	t = 874,878,882,886,890,894,898,902,906,910,914,918,922,926,930,934,938,942,946,950,954,958,962,966,970,974,978,982,986,990,994,998,1002,1006,1010,1014,1018	[119]
BF264-15Dau Wild type cells, YEP medium	t = 30,38,46,54,62,70,78,86,94, 102,110,118,126,134,142,150,158,166,174,182,190,198,206,214,222,230,238,246,254,262	[120]
BF264-15Dau D CLB1 cells, YEP medium	t = 30,38,46,54,62,70,78,86,94, 102,110,118,126,134,142,150,158,166,174,182,190,198,206,214,222,230,238,246,254,262	[120]

Table 2.S6. *S. cerevisiae* data set for DREAM5 lag analysis

Cluster ID	Gene Ontology	Significance (Corrected P-Value)	# of Genes in GO	Size of GO Category
0	glycogen metabolic process	0.008232665	4	59
2	oxidation-reduction process	2.07E-22	87	223
3	transcription, DNA-templated	0.032096568	23	93
4	D-gluconate metabolic process	2.80E-12	7	13
5	carbohydrate transport	9.24E-29	53	299
5	carbohydrate metabolic process	3.18E-25	59	299
5	cytoplasm	0.034127985	94	299
6	iron ion homeostasis	8.10E-23	21	111
6	ion transport	4.93E-09	22	111
6	transport	6.25E-06	46	111
7	sulfate assimilation	6.26E-09	7	28
7	sulfur compound metabolic process	2.51E-08	7	28
8	oxidation-reduction process	0.016750762	22	68
9	lipid metabolic process	7.45E-20	16	22
10	cellular amino acid biosynthetic process	3.55E-08	19	95
11	membrane	2.92E-07	20	173
12	cellular response to DNA damage stimulus	6.90E-17	30	61
14	phosphate ion transport	1.32E-10	8	54
16	glyoxylate catabolic process	0.001176962	3	9
17	response to arsenic-containing substance	1.40E-05	3	3
19	biotin biosynthetic process	3.46E-12	6	6
20	cyanate catabolic process	1.40E-05	3	4
21	glycerol metabolic process	0.00127654	3	3
23	aromatic compound catabolic process	1.91E-08	5	6
25	potassium-transporting ATPase activity	1.83E-08	4	5
27	NAD biosynthetic process	1.15E-06	4	5
28	phage shock	0.000280394	3	6
30	thiamine transport	7.01E-05	3	6
31	cellular amino acid biosynthetic process	2.49E-12	13	20
32	nucleobase-containing small molecule metabolic process	0.013445757	2	3
35	zinc II ion transport	0.003916298	3	6

Table 2.S7. Gene ontological analysis of *S. cerevisiae* subnetworks

CHAPTER 3

Hybrid analysis of gene dynamics predicts context specific expression and offers regulatory insights

This work was published with Neda Bagheri in *Bioinformatics*[121]

3.1. Abstract

Motivation: To understand the regulatory pathways underlying diseases, studies often investigate the differential gene expression between genetically or chemically differing cell populations. Differential expression analysis identifies global changes in transcription and enables the inference of functional roles of applied perturbations. This approach has transformed the discovery of genetic drivers of disease and possible therapies. However, differential expression analysis does not provide quantitative predictions of gene expression in untested conditions. We present a hybrid approach, termed DiffExPy, that uniquely combines discrete, differential expression analysis with *in silico* differential equation simulations to yield accurate, quantitative predictions of gene expression from time-series data.

Results: To demonstrate the distinct insight provided by DiffExpy, we applied it to published, *in vitro*, time-series RNA-seq data from several genetic *PI3K/PTEN* variants of MCF10a cells stimulated with epidermal growth factor (EGF). DiffExPy proposed ensembles of several minimal differential equation systems for each differentially expressed gene. These systems provide quantitative models of expression for several previously uncharacterized genes and uncover new regulation by the *PI3K/PTEN* pathways. We

validated model predictions on expression data from conditions that were not used for model training. Our discrete, differential expression analysis also identified *SUZ12* and *FOXA1* as possible regulators of specific groups of genes that exhibit late changes in expression. Our work reveals how DiffExPy generates quantitatively predictive models with testable, biological hypotheses from time-series expression data.

Availability: DiffExPy is available on GitHub *(https://github.com/bagherilab/diffexpy)*.

Contact: n-bagheri@northwestern.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

3.2. Introduction

Aberrant regulation of gene expression is frequently associated with diseases; thus, changes to gene expression serve as key proxies to infer cell state[122]. Differential gene expression analysis quantifies changes in gene expression between cell states. Expression is compared between genetically different cells, cells exposed to different exogenous treatments—such as small molecules, proteins, temperatures, or other environmental cues—or a combination of several treatments. Each gene in the analysis is then categorized as a differentially expressed gene (DEG) or not. This categorization is often based on the magnitude of the log fold change (LFC) of its expression between experimental conditions and by an adjusted p-value. DEGs are often split into groups of genes that are overexpressed or underexpressed[123,124]. Finding enriched Gene Ontology (GO) terms or pathways associated with the DEG can elucidate the functional role of the experimental condition[22,24].

Measuring and analyzing the dynamics of gene expression are also critical to understanding responses involved in DNA repair, development, and circadian rhythms[17,125,126].

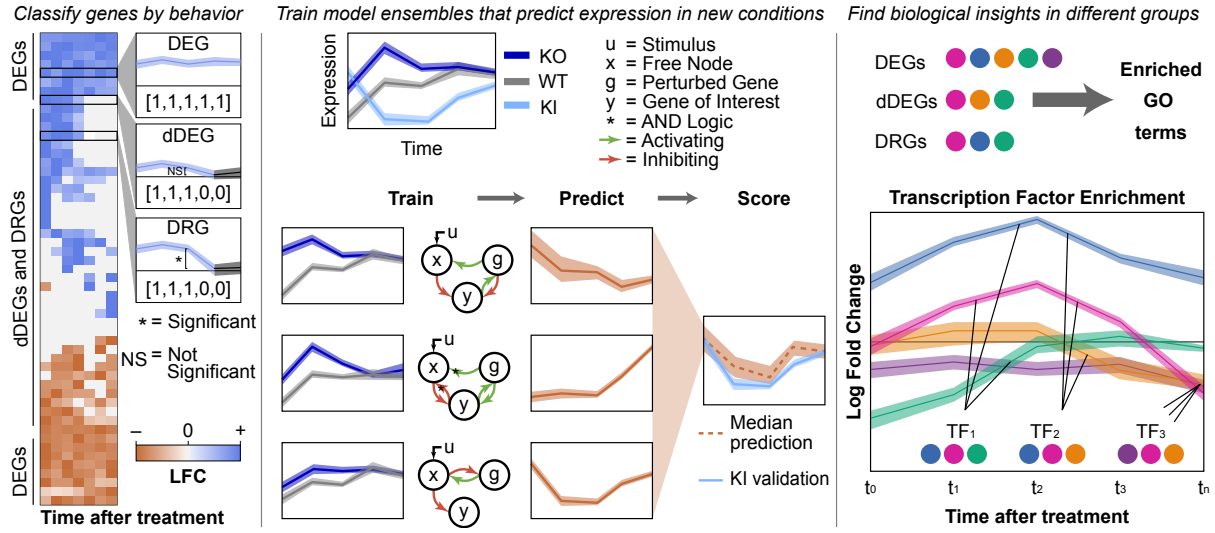


Figure 3.1. Overview of DiffExPy analysis. (Left) Genes are categorized as differentially expressed genes (DEGs), dynamic DEGs (dDEGs), or differentially responding genes (DRGs) from time-course gene expression data. Discrete responses (in brackets) are determined for each contrast. (Center) Stochastic differential equation systems that match the dDEG gene profiles are selected from a library of possible models and combined into an ensemble model. The ensembles can predict gene expression behavior in new, untested conditions. (Right) Biological insights are gained by associating GO terms with gene classifications and associating TFs with groups of genes that share discrete differential behavior.

A typical time-series, gene expression experiment compares expression between experimental conditions over several time points[17,127]. Many algorithms that identify DEGs from time-series data exist, but these algorithms focus on DEG identification for subsequent enrichment analyses[128]. Other algorithms use time-series expression data to infer the structure of gene regulatory networks[48,86,129], or attempt to identify transcription factors (TFs) that explain changes in time-series gene expression, by pairing the expression data with ChIP-seq data[127,130].

A limitation of existing differential expression analyses—both of static and time-series data—is that they do not propose quantitative models of a gene's expression that can be tested in new experimental conditions. For example, if a gene is overexpressed in cells treated with a particular drug (compared to untreated cells), existing analyses cannot

predict if that gene will be overexpressed, underexpressed, or unchanged when a different drug is applied. Researchers can only infer how the regulation might occur and qualitatively predict how expression will differ in untested contexts.

Distinct from statistical enrichment approaches, differential equation models aim to use mechanistic information to describe how species, such as genes or proteins, interact and are well-suited to quantitatively predict gene expression in untrained conditions. However, designing and fitting differential equation parameters requires sufficient data; therefore, such models only exist for a few well-studied systems[64–67]. Genetic and sparse regression algorithms can generate differential equation models directly from data, but current gene expression technologies cannot produce the highly sampled, low-noise data these algorithms require[69,71].

Data-driven methods to generate models that can quantitatively predict gene expression are currently limited. Network inference methods generate genome-scale models, however to predict the expression of any one gene requires knowing the expression of several others[48,55,93]. Other methods explicitly fit expression to a time variable, which ignores the molecular contexts driving expression[17]. To fill this gap, we present **Diff**erential **Ex**pression in **Py**thon (DiffExPy), a framework that uses time-series expression data to create dynamical-systems models of gene expression.

DiffExPy first determines a discrete response from the expression of each gene in the time series based on the sign and significance of the gene's LFC between conditions at each time point. Next, DiffExPy simulates time-series expression data from a library of minimal stochastic differential equation (SDE) systems that mimic the experimental conditions. Then, the discrete response of a gene is matched to models in the simulation library to train an ensemble model. This trained model can predict that gene's expression

in new conditions. DiffExPy also clusters genes by discrete response, and infers the timing of regulatory events by associating these gene groups with transcription factors and GO terms (Fig. 3.1).

We demonstrate the efficacy of DiffExPy on publicly available RNA-seq data from the GeneExpressionOmnibus(GEO) repository, accession number GSE69822[25]. Previous analysis of this data set further elucidated the transcriptional roles of phosphoinositide 3-kinase (*PI3K*) and phosphatase and tensin homolog (*PTEN*), which respectively phosphorylate and dephosphorylate phosphatidylinositidol-4,5-bisphosphate (PIP2) to and from phosphatidylinositidol-3,4,5-trisphosphate (PIP3). PIP3 regulates many downstream pathways, most notably the *AKT* pathway[25]. For our analyses, we use data from the wildtype (WT), *PTEN* knockout (*PTEN* KO), A66 treated cells, and *PI3K* knockin (PIK3CA H1047R) conditions. A66 inhibits the p110α *PIK3CA* and we refer to it as the inhibited condition ($PI3K^{\text{inh}}$). The histidine-to-arginine substitution makes PIK3CA constitutively active, and we refer to it as the knockin (*PI3K* KI) condition. In the original study, expression was measured from MCF10a cells, a commonly used human breast epithelial cell line, stimulated with epidermal growth factor (EGF) using RNA-seq in three replicates at 0, 15, 40, 90, 180, and 300 minutes after EGF stimulation[25].

Using the differential-expression data between the $PI3K^{\text{inh}}$ and WT conditions, we train ensemble models for several genes. We validate the expression predictions for each gene using the *PI3K* KI time-series data and provide a straightforward approach to rank the confidence of each trained ensemble. The ensembles vary in size and consist of minimal SDE systems with different connectivitivies. We highlight results of three genes known to interact with the *PI3K* pathway that currently lack quantitative models for their expression. In doing so, we demonstrate how the trained ensemble models provide simple

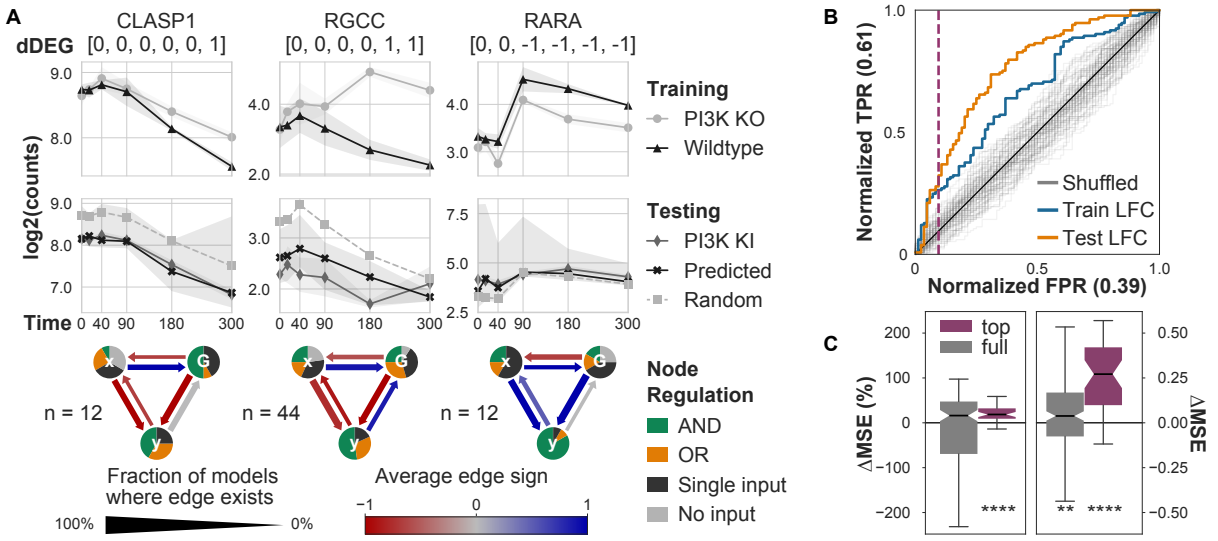


Figure 3.2. Ensembles of minimal SDE systems trained on *PI3K*$^{\text{inh}}$ and WT data accurately predict expression in untrained *PI3K* KI condition. (A) Three examples of dDEGs with matching dynamical models. (A, top row) Normalized gene expression for each gene in *PI3K*$^{\text{inh}}$ and WT used to train the models. The discrete response of the pairwise contrasts (in brackets) for each dDEG show when differential expression occurs. (A, middle row) Normalized gene expression for model predictions, null model predictions, and true *PI3K* KI expression. Trained and null model predictions are median values, and the filled regions show the 83% CI of the median. (A, bottom row) Network diagrams summarize the ensemble models matched to each gene in the training condition. (B) AUROC curve plot of different methods for sorting gene predictions. Sorting by mean LFC between the training conditions places more accurate predictions at the top of the list. A threshold for selecting more accurate predictions (purple, dashed line) is calculated using the elbow rule of the sorted mean LFC values in the training condition (Fig. 3.S7). (C) Box plots show the normalized and absolute difference in MSE of the trained models compared to the paired random models for all genes. The top set of genes (purple) were determined by the elbow rule and are significantly more likely to generate more accurate predictions. p-values were calculated using the Wilcoxon signed-rank test. ** p<0.01, **** p<0.0001.

starting models for less-studied genes. We also use the discrete response calculated by DiffExPy to identify the timing of regulation by suppressor of zeste 12 (*SUZ12*) and forkhead box A1 (*FOXA1*) on their target genes.

DiffExPy is distinct from the status quo in generating dynamical system models *de novo* for many genes that were not previously characterized. Currently, DiffExPy is limited in that it constructs small models based on time-series data from individual

perturbations. However, DiffExPy is readily extensible and can be adapted to other differential-expression packages, model assumptions, and genomic data. For example, future improvements to DiffExPy could be made to incorporate multiple perturbations, additional omics data types and prior knowledge. Our work provides a foundation on which more complex models of gene expression can be developed.

3.3. Materials and Methods

Using time-series RNA-seq data, DiffExPy sorts genes according to their discrete, dynamic, differential gene expression profiles (Figs. 3.1 and 3.S1). Each gene's discrete profile is used to train an ensemble model of minimal stochastic differential equation (SDE) systems that predict expression in new conditions (Fig. 3.1). DiffExPy also associates GO terms with resulting groups, which suggest functional roles for the genes in each distinct cluster. Finally, DiffExPy associates TFs with genes that exhibit similar responses at specific times (Fig. 3.1). Overall, DiffExPy identifies (i) minimal dynamical systems models that accurately predict gene expression dynamics in untrained conditions, (ii) specific GO terms associated with classes of expression dynamics, and (iii) specific TFs associated with genes with similar expression responses.

3.3.1. DiffExPy assigns discrete differential responses

To match gene expression responses to dynamical systems models, DiffExPy first calculates discrete responses from LFC contrasts generated by differential expression analysis (Fig. 3.S1) using the package *limma*[44]. A contrast is defined as a comparison of expression between conditions, time points, or both. The discrete response is derived from the LFC value for a contrast, which can be positive (+1), negative (-1), or not significant (0). We assign gene labels based on discretized LFC values (Figs. 3.1 and 3.S1) as differentially

expressed genes (DEGs), dynamic DEGs (dDEGs), and differentially responding genes (DRGs).

DEGs are genes that are differentially expressed between conditions at one or more time points after the treatment, based on an F-test. dDEGs define the subset of DEGs that exhibit dynamic, or variable, differential expression across time. For instance, an expression profile need not be differentially expressed at time 0, but it can become differentially expressed at a later time point. DRGs contain at least one time point in which the LFC is significantly different from the LFC either at the previous time point (*i.e.*, $LFC_t \neq LFC_{t-1}$) or from the time the treatment is applied (*i.e.*, $LFC_t \neq LFC_0$), where significance is determined using an F-test. Classification of a gene as a dDEG or DRG is not mutually exclusive, and by definition a dDEG or DRG is also a DEG.

3.3.1.1. Definition of gene expression contrasts. A gene expression contrast compares the distribution of expression values of a gene between samples[44]. Using time-series data, the basic set of values used in a contrast for gene i, given condition c, with R replicates, and at time t is defined as:

$$\vec{g}^{\,t}_{i|c} = [g_i^{1,t}, g_i^{2,t} \ldots, g_i^{R,t}] \tag{3.1}$$

The LFC is calculated as the ratio of the mean log_2 expression value between the conditions:

$$l_i^t = \frac{\langle \vec{g}^{\,t}_{i|exp} \rangle}{\langle \vec{g}^{\,t}_{i|ctrl} \rangle} \tag{3.2}$$

where *exp* is the experimental condition and *ctrl* is the control condition. By convention, the control condition is in the denominator, so positive LFC values correspond to overexpression in the experimental condition. For each contrast, a corresponding p-value is

calculated. When multiple contrasts are made, an overall significance level is also calculated using an F-test. Significance levels are corrected for multiple hypothesis testing[44]. Differential expression calculations between genes are linearly independent and are easily extended to matrix form.

DiffExPy departs from the status quo by using time-series data to create more complex contrasts. Pairwise (PW) contrasts compare expression between experimental conditions at each time points. Time-series (TS) contrasts compare expression between a time point and the previous time point. Autoregressive (AR) contrasts compare expression between a time point and the time point before the treatment was applied. A detailed description of these and other combinations of contrasts (PW-TS and PW-AR) is available in *Supplementary Information* (Fig. 3.S1).

3.3.1.2. Discrete expression responses. To facilitate downstream analyses, DiffExPy calculates a discrete response for each gene based on the p-values and signs of LFC for the individual contrasts. If the p-value for a contrast is above the user-specified threshold, the LFC is not considered significant and is set to zero. The discrete response for gene i is defined as $\vec{d}_{i,x} = [d(l)] \; \forall \; l \in \vec{l}_{i,x}$, where x is one of the set $\{PW, TS, AR,$ PW-TS, or PW-AR$\}$. The discrete values are calculated using the signs of the LFC values as follows:

$$d(l) = \begin{cases} 1 & sign(l) > 0 \; and \; p(l) < p_{cut} \\ 0 & p(l) \geq p_{cut} \\ -1 & sign(l) < 0 \; and \; p(l) < p_{cut} \end{cases}, \tag{3.3}$$

where $p(l)$ is the adjusted p-value of the LFC for the contrast and p_{cut} is the significance threshold. For a time series of T time points, each discrete response has 3^{T-1} possible

clusters—except for $\vec{d}_{PW}$, which has 3^T. We did not filter LFC values by magnitude, but the option is provided in DiffExPy. We used a p-value threshold of 0.05 for all of our tests. A lower p-value cutoff could result in discrete responses with more zero values.

3.3.2. Training predictive models of gene expression

DiffExPy uses GeneNetWeaver (GNW) to generate minimal differential equation models for unique, three-node networks and to carry out stochastic simulations. GNW models include a protein and mRNA component[61]. Each model is an abstract representation of the flow of information that might regulate a gene's expression and should not be used to identify specific regulation between genes. To mimic the experimental data, we used DiffExPy to conduct three independent, stochastic runs of each model and sampled each model at the same time points as the RNA-seq data. Microarray-like measurement noise was added to the data, and values were normalized between 0 and 1. A complete description of model generation is available in the *Supplementary Information*.

Simulations were conducted under three genetic conditions: wildtype, knockout, and knockin. It is important to note that the identity of the perturbed gene does not need to be known to gain information from DiffExPy. If a treatment with an unknown target is applied, DiffExPy can still provide insight into possible motifs of which a gene is a part that results in the observed expression behavior.

3.3.2.1. Matching models to genes. After simulation, each SDE model had time-series data mimicking the experimental data. We conducted the same discrete clustering process on the simulated data. Each gene was matched to all SDE models with the same discrete response as the gene $match_i = \{m \in M | l_i^t = l_m^t \ \forall \ t \in T\}$, where M is the set

of all models in the simulation library and T is the number of time points in the discrete response.

3.3.3. Model predictions

Each SDE model matched by DiffExPy can then simulate time-series data under different conditions to generate predicted LFC values at each time point, represented as a T-length vector $\hat{l}_k = [\hat{l}^1, \ldots, \hat{l}^T]$, where k is the index of the matched SDE system and T is the number of time points in the predicted time series. The time series of the control condition provides an internal control for predicting the response.

For our predictions, we applied a simulated knockin of *PI3K* to each of the trained SDE models as this matches the *PI3K* KI perturbation that was applied in the experimental data. Importantly, predictions to match different treatment strategies can also be made—such as targeting multiple nodes in the SDE models, inhibiting interactions between SDE nodes, or changing the forcing function.

We found no feature of the individual models that correlated with their prediction error. We therefore created an ensemble prediction by using the median predicted LFC. For each gene i, the predicted LFC is calculated as the median LFC of all matched SDE predictions for each time t:

$$\hat{l}_i^t = median([\hat{l}_m^t \; \forall \; m \in match_i]). \tag{3.4}$$

The simulated model predictions in log_2 expression space are $\hat{y}_i^t = b_i^t + \hat{l}_i^t$, where b_i^t is the log_2 expression of the control condition at time t, $\hat{l}_i^t$ is the model predicted LFC, and $\hat{y}_i^t$ is the predicted log_2 expression of the experimental condition.

3.3.3.1. Scoring prediction accuracy. We validated the accuracy of the quantitative predictions by calculating the error between *PI3K* KI expression of a gene and its corresponding model's prediction. We define accuracy as the mean squared error (MSE) between a model's average LFC (of 3 stochastic runs) and the true LFC. MSE values range from 0 to ∞, where smaller values indicate the prediction is closer to the true LFC value. We know of no existing, data-driven method that generates models capable of quantitative, time-series, gene expression prediction to provide an appropriate basis for comparison. Thus, we used the selection of a random model from our library as the null model comparison.

3.4. Results

3.4.1. Many previously uncharacterized genes are matched to ensemble models

We used the discrete response profiles to match each gene to an ensemble of three-node SDE models that each share similar dynamics upon simulation. Our library consists of 2,172 uniquely structured SDE models. We trained the models using the $PI3K^{\text{inh}}$ and WT data, and we used the *PI3K* KI data as test data to validate the predictions. Simulations for each network model were created to match the *PI3K* genetic condition and EGF stimulation. The simulated data is sampled at the same time points used in the experiments. Details of the library creation are provided in the *Supplementary Information*.

As the $PI3K^{inh}$ does not affect the expression of many genes[25], DiffExPy only identifies 223 dDEGs from the differential expression analysis of the $PI3K^{inh}$ and WT data. There is a many-to-many match between the dDEGs and the possible three-node SDE systems.

217 of the dDEGs were matched to at least one network model. Of the 217 matched genes, few are well-studied; there is sparse information about their functional role. We identified just 9 genes whose paralogs were likely to exist in current computational models of well-studied signaling pathways. These included genes in the *MAPK*, *JAK/STAT*, and *PI3K/AKT/mTOR* pathways[64–67].

3.4.2. Ensemble models highlight possible dynamical systems from which to build more detailed models

Each independent model in the ensemble suggests a possible SDE system whose simulations match the qualitative features of the experimentally measured expression. Specifically, each SDE system represents how the gene of interest (node y) might interact with the perturbed gene (node G) and the rest of the genome (node x). In this experiment, G represents *PI3K* as it was the knocked-out, knocked-in, or inhibited gene. Summaries of the models that match each gene and create the quantitative predictions reveal possible regulatory interactions that result in the observed dynamics (Fig. 3.2A). We highlight results of trained models for three genes that exhibit different discrete responses and predictive accuracy: cytoplasmic linker associated protein 1 (*CLASP1*), regulator of cell cycle (*RGCC*), and retinoic acid receptor alpha (*RARA*). *CLASP1* and *RARA* were previously shown to interact with components of the *PI3K/AKT* pathway[131,132].

Models for *CLASP1* and *RGCC* primarily contain inhibition by both x and G, whereas x and G appear as activators of *RARA*. Furthermore, in almost all models matched to *RARA*, both x and G must be present to activate *RARA*. The *RGCC* models often contain activation of G by *RGCC*. Conversely, models of *CLASP1* and *RARA* often exhibit feedback on G but are not consistently activating or inhibiting. Overall, these models

suggest modes of regulation between *PI3K* and *CLASP1, RGCC*, and *RARA*, and predict gene responses to future conditions and treatments. These models also provide a basis from which more detailed models can be developed.

3.4.3. DiffExPy ranking sorts models by predictive accuracy

Because every ensemble model is not equally predictive of its respective gene, we searched for a metric to rank model predictions. We find that the mean absolute LFC between the *PI3K* KI and WT conditions correlates with improved model accuracy (Spearman's ρ=0.636, p=5.4e-26, Fig. 3.S2). Ordering gene predictions by the mean absolute LFC places genes with lower MSE at the top of the list. Treating genes with positive ΔMSE (*i.e.*, lower error than random) as positive classifications, we can assess the area under the receiver operator characteristic (AUROC) and area under precision recall (AUPR) curves. The AUROC for this ranking is 0.76 and the AUPR is 0.78, both of which are significantly greater than expected from a random ordering (Figs. 3.2B and 3.S3).

Since future experiments will not always include validation data, we sought a proxy for model confidence. We find that the mean absolute LFC between the *PI3K*$^{\text{inh}}$ and WT correlates with the mean absolute LFC between the *PI3K* KI and WT (Spearman's ρ=0.684, p=2.7e-31, Fig. 3.S4). The mean absolute LFC between the *PI3K*$^{\text{inh}}$ and WT also correlates with improved model accuracy (Spearman's ρ=0.418, p=1.4e-10, Fig. 3.S5). This ranking yields an AUROC of 0.66 and an AUPR of 0.73, which are slightly lower, but still significantly better than random (Fig. 3.2B, Fig. 3.S3).

We believe this sorting is intuitive. A gene with a greater effect size in the transcriptional response provides more information during training, which results in better matched models. On average, a random model predicts no LFC between WT and another

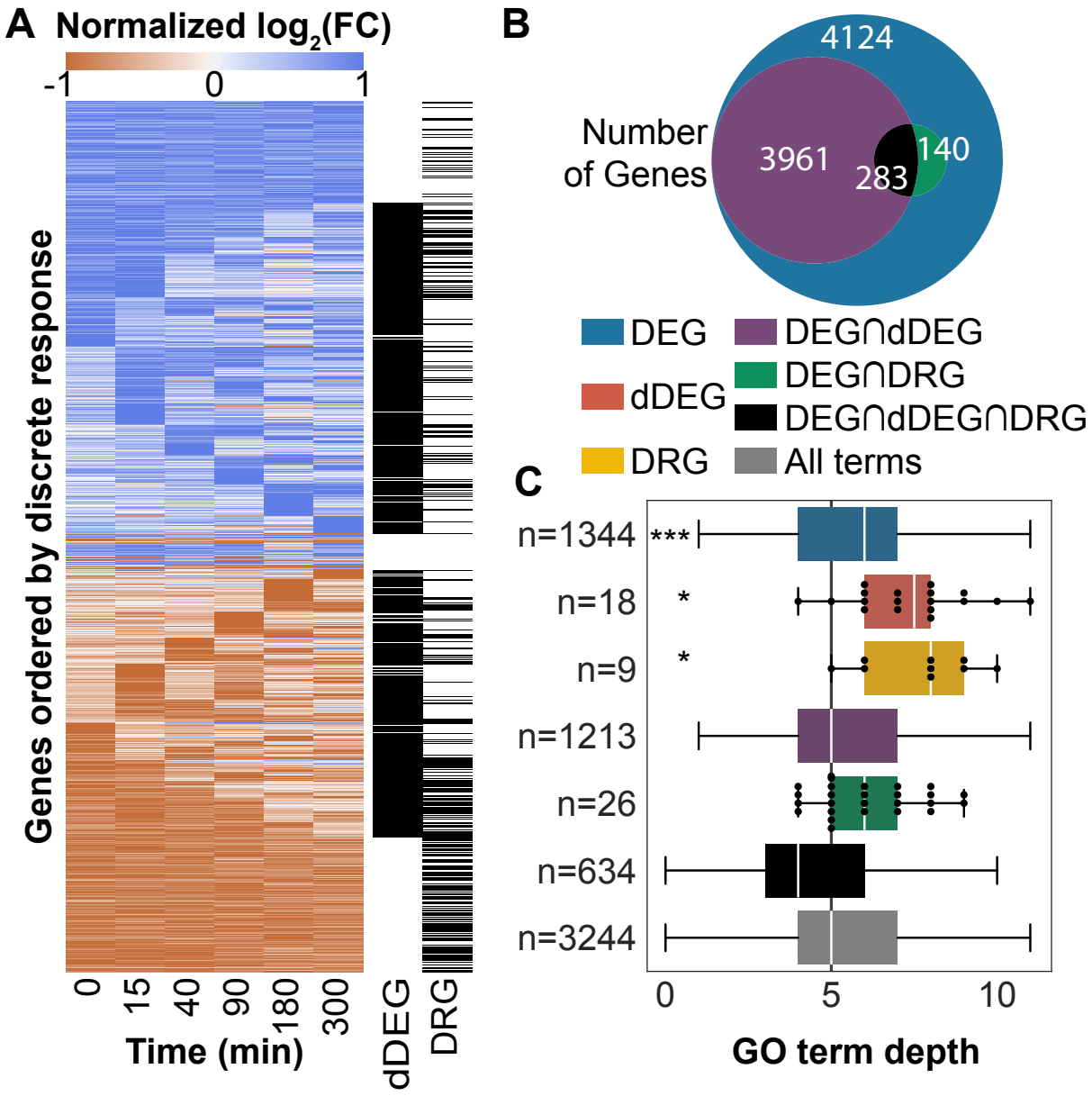


Figure 3.3. Summary of gene classifications comparing *PTEN* KO to WT. (A) Heatmap of row normalized LFC for all DEGs. Genes that are classified as dDEG or DRG are also labeled. (B) Overlap of genes that are classified as DEG, dDEG, DRG, or some combination. By definition, all dDEGs and DRGs must also be DEGs. (C) Comparison of distributions of GO term depths uniquely associated with intersections of gene sets. Even though no genes are classified as only dDEG or DRG, these gene sets associate with unique, specific GO terms not found in the dDEG set. The number of unique terms associated with each set, n, is shown. p-values were calculated using a discrete KS test. * $p < 0.05$, *** $p < 0.001$.

condition for a given gene, so a trained model prediction should be more accurate (Fig. 3.S6).

3.4.4. Top-ranked genes offer accurate predictions

Overall the predictions for the dDEGs have a median MSE that is 0.027 (p=0.018) lower than random (ΔMSE). However, after ranking genes by mean $PI3K^{\text{inh}}$-WT LFC, we applied an elbow rule to select the top 40 genes. Our results indicate that the top-ranked genes have significantly more accurate predictions than random models. The top genes have a median ΔMSE of 0.523 (p=1.45e-6) and a %MSE of 33.3% (p=1.74e-5, Fig. 3.2C). The elbow rule gives an empirical threshold to select the top predictive genes (Fig. 3.S7).

3.4.5. Gene classifications from discrete responses associate with specific GO terms

To demonstrate the high-level biological insights gained from the discrete responses, we present results of classifying genes from their discrete responses comparing the *PTEN* KO to WT time-series expression data. We identify 8,508 DEGs, of which 3,961 are classified as dDEGs, 140 as DRGs, and 283 as all three (Fig. 3.3B). We performed GO term enrichment analysis on each of these non-mutually exclusive gene classifications. Enriched GO terms were grouped by the exclusive set to which they belonged. Thus, genes can have multiple labels, but GO terms can only be associated with one group. For example, a GO term associated with both the sets of DEGs and dDEGs—which have many overlapping genes—would be assigned to the DEGs∩dDEGs group, whereas a GO term only associated with the set of dDEGs would be assigned to the dDEG group (Figs. 3.3B and C).

All terms were called from the same directed acyclic graph (DAG). Term depth quantifies the level in the GO hierarchy, and it is used as proxy for term specificity. Even though no genes are categorized exclusively as dDEGs or DRGs, there are very specific terms associated only with these groups (Fig. 3.3C). Results of the gene classifications for the *PI3K* KI and *PI3K*$^{\text{inh}}$ compared to WT are provided in the *Supplementary Information* (Figs. 3.S8 and 3.S9). Specific gene clusters with similar discrete responses may also be used for GO enrichment analysis, but we next focus on using them for TF enrichment.

3.4.6. TF enrichment suggests regulators of gene expression

Similar to GO term enrichment analysis, we calculate TF enrichment for gene clusters. A group of genes enriched for association with a particular TF may indicate that the TF is responsible for the observed change in expression. Existing methods, such as weighted gene co-expression network analysis (WGCNA) and dynamic regulatory events miner (DREM), perform clustering of gene profiles for subsequent GO and TF enrichment analysis[40,127,133]. In contrast, DiffExPy uses the discrete LFC values (0,1,-1) to generate default clusters. This discretization enables grouping genes in various ways that suggest different types of coregulation by a shared TF.

For example, the set of all DRGs is enriched for association with 52 TFs (adjusted $p < 0.05$). This set includes supressor of zeste 12 (*SUZ12*) and forkhead box A1 (*FOXA1*), which were not identified in the original study[25]. *SUZ12* is a zinc finger protein and a component of the polycomb repressive complex 2 (PRC2). PRC2 has histone methylation activity, yet its regulatory role in cell fate is uncertain[134,135]. *FOXA1* is an important TF in breast and prostate cancers and is known to be a target of both MAPK and AKT[136,137].

Using the temporal information inherent to a time-series data set, we can identify when, and how, the regulation by these factors occurs. For example, we identify the set of 41 genes that have lower LFC at 300 min than at 0 min, which are enriched for association with *SUZ12* (Fig. 3.4A). 13 of the 41 genes are known to be associated with *SUZ12*, and the enrichment suggests that changes in expression for this group are regulated by *SUZ12*. Interestingly, there is no identifiable change in *SUZ12* expression (Fig. 3.4A), indicating that downstream gene regulation by *SUZ12* might depend on post-trancriptional changes, such as sumoylation[138].

We also find *FOXA1* to be associated with genes that show an increase in LFC between 90-180 min. A natural hypothesis is that this group of 79 genes all show the same change in expression at these later time points because they share a common regulator, *FOXA1* (Fig. 3.4B). In contrast to *SUZ12*, *FOXA1* exhibits a distinct differential response beginning 90 min after the EGF stimulus. Several of the genes that have the described behavior and are associated with *FOXA1* show a similar qualitative differential response to EGF as *FOXA1*. These results suggest that *FOXA1* might regulate the expression of these genes, as well as others in the set, in response to EGF stimulation. Additionally, each of these genes, including *FOXA1*, is classified as a DRG, further supporting the hypothesis that *PTEN* is required for proper expression of these genes in response to EGF stimulation.

SUZ12 and *FOXA1* are not the only TFs associated with discrete response clusters. Instead, these examples demonstrate two possible ways the discrete analysis might identify enriched regulators for groups with different response behaviors. The timing of the expression behavior creates strong, testable hypotheses for the inferred regulators. Additional enrichment results, and a complete discussion of the enrichment methods, is are provided in the *Supplementary Information*.

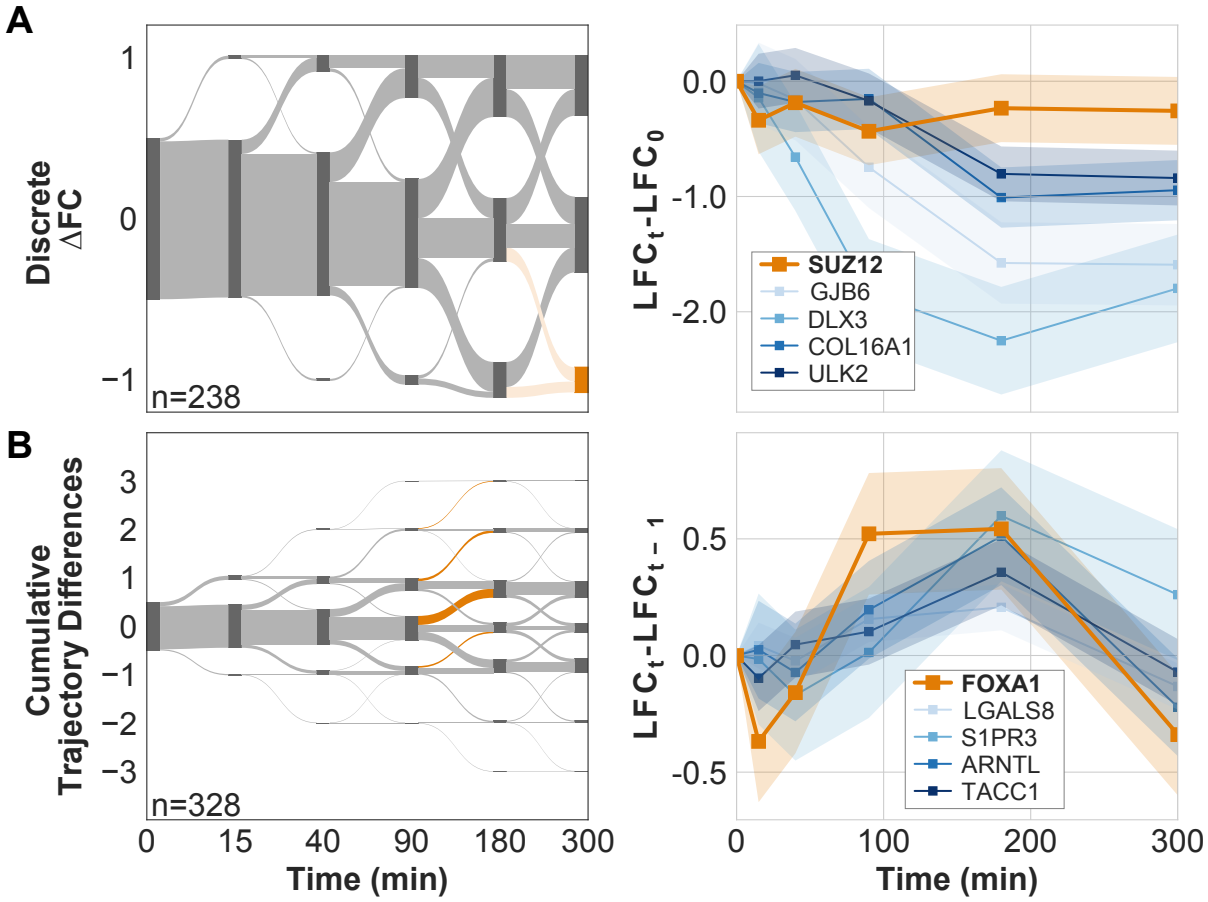


Figure 3.4. Specific timing of changes in gene expression identifies possible regulators. (A, left) Sankey plot of discrete FC between *PTEN* KO and WT compared to FC before the stimulus is applied shows when DRGs respond to the stimulus. Nodes (dark grey) are scaled by the number of genes in that state. Edges (light grey) show the fraction of genes moving from one state to another between time points. The node and edges highlighted in orange show the set of genes associated with *SUZ12*. (A, right). LFC values relative to LFC before stimulus for *SUZ12* and several genes associated with *SUZ12*. (B, left) Sankey plot of the cumulative differences in *PTEN* KO slope and WT slopes. Segments highlighted in orange show genes whose KO expression increases more than their WT expression between either 90-180 min or 180-300 min. These genes are enriched for association with *FOXA1*. (B, right) LFC values relative to LFC at the previous time point for *FOXA1* and several associated genes.

3.5. Discussion

The characterization of less-studied genes is fundamental to understanding cellular responses in diverse environmental contexts[139]. In this study, we presented DiffExPy, an

analysis framework that calculates discrete differential expression responses and trains dynamical systems models for many quantitatively uncharacterized genes.

We demonstrated how, for each matched gene, DiffExPy prototypes quantitative models that offer accurate predictions of gene expression in untrained conditions. We also validated the DiffExPy model predictions of each gene's expression in the untrained *PI3K* KI condition (Fig. 3.2). Our results suggest that *PI3K* inhibits expression of both *CLASP1* and *RGCC*, often in conjunction with additional factors. These *de novo* results are supported by previous experiments. *CLASP1* was shown to interact with proteins affected by *PI3K* activity[131]. *RGCC* was also demonstrated to have several regulatory roles in the *PI3K* pathway[140]. Discrepancies with other data can refine the models and our understanding of each gene's regulation. For example, the summary of *RARA* suggests that for the observed response, *PI3K* is a positive regulator of *RARA*. This result is surprising because *PI3K* activates AKT, which was shown to subsequently inhibit *RARA*[132]. One explanation is that an unknown activating path between *PI3K* and *RARA* exists.

We also demonstrated how DiffExPy associates groups of similar, discrete gene expression responses with TFs, such as *SUZ12* and *FOXA1* (Fig. 3.4). Though *SUZ12* expression does not differ between the $PI3K^{\text{inh}}$ and WT conditions, several known and possibly new targets of *SUZ12* exhibit a differential response to EGF stimulus. These observations might suggest that regulation by *SUZ12* results from post-translational modification[138]. Conversely, *FOXA1* and many of its targets exhibit a differential response, which is consistent with previously studied interactions between *FOXA1* and the AKT pathway[136,137]. Finally, we show how each gene classification associates with GO terms that enable a unique understanding of regulatory functions.

DiffExPy was formulated to create predictive models for many genes with varied dynamic expression responses. We limited our model library to three-node gene regulatory networks without self-edges (Fig. 3.S10). As such, a suitable model match might not exist in the library for each gene. In our analysis, a small fraction of genes (6 of the 223 dDEGs, or less than 3%) did not match to a suitable model. The absence of matches might be attributed to the limited scope of SDE models in the library, limiting possible gene expression dynamics. The library could be expanded to include four-node networks, which might be capable of simulating more qualitatively-diverse expression dynamics. Expanding the library might be computationally expensive and require optimizing the library generation step, simulation, and matching. Additionally, the current SDE models could be simulated with different kinetic assumptions, though the accuracy of these assumptions would need to be validated. Finally, because our training and testing perturbation affect the same gene, the ranking of the results might not hold for all predictions. We also only focused on comparisons between pairs of experimental conditions (*i.e.*, only one node is perturbed). Integration of multiple experimental conditions simultaneously might yield more predictive models but training the models more difficult.

A complete understanding of cellular regulation cannot be gained only from transcriptomics. Epigenomic data from ChIP-seq, ATAC-seq, etc, can provide additional information to increase the mechanistic specificity of the models by identifying direct regulators and chromatin accessibility. Currently, TF enrichment is calculated using associations derived from ENCODE[127]. Direct measurements of TF association or chromatin accessibility, during the same time course, could be directly integrated into the existing framework. Unfortunately, this data was not available for our analysis and might currently be cost prohibitive.

Overall, few genes have detailed biochemical models that quantitatively predict their behavior in diverse conditions. Characterizing how less-studied genes are regulated in multiple contexts will improve our understanding and treatment of disease. The models generated by DiffExPy provide systematic, reliable starting points for quantitative models of regulation based on time-series, differential-expression data.

3.6. Funding

This research was supported, in part, by a Nicholson Fellowship and Biotechnology Training Program Fellowship (to J.D.F), NSF CAREER Award (CBET-1653315 to N.B.), and the McCormick School of Engineering.

Supplementary Materials and Methods

Experimental data set

All code used in the analysis of gene expression, and all *in silico* data is available on GitHub (*https://github.com/bagherilab/diffexpy*). The *in vitro* data used in this study was previously published and is available in the Gene Of Expression (GEO) repository, with accession number GSE69822[25].

Gene expression data representation

The measurement of expression for gene i with R replicates is defined by the R-length vector:

$$\vec{g_i} = [g_i^1, g_i^2 \ldots, g_i^R]^\top \tag{3.5}$$

When gene expression is measured under different conditions for differential expression analysis, the measurement of expression for gene i, given condition c, with R replicates is expanded to the following R-length vector:

$$\vec{g}_{i|c} = [g_i^1, g_i^2 \ldots, g_i^R]^\top \tag{3.6}$$

A gene expression experiment that measures N genes is represented as an $R \times N$ matrix:

$$\begin{aligned} \mathbf{E_c} &= [\vec{g}_{1|c}, \ldots, \vec{g}_{N|c}] \\ &= \begin{bmatrix} g_1^1 & \cdots & g_N^1 \\ \vdots & \ddots & \vdots \\ g_1^R & \cdots & g_N^R \end{bmatrix} \end{aligned} \tag{3.7}$$

Time-series gene expression data representation

Time-series gene expression data is represented by vertically concatenating gene expression matrices for T time points into a $(T * R) \times N$ matrix as follows:

$$\begin{aligned} \mathbf{T_c} &= [\mathbf{E_c^1}, \ldots, \mathbf{E_c^T}]^\top \\ &= \begin{bmatrix} g_1^{1,1} & \cdots & g_N^{1,1} \\ \vdots & \ddots & \vdots \\ g_1^{R,1} & \cdots & g_N^{R,1} \\ \vdots & \ddots & \vdots \\ g_1^{R,T} & \cdots & g_N^{R,T} \end{bmatrix} \end{aligned} \tag{3.8}$$

Here, each gene is sampled consistently. While the sampled time points do not need to be evenly spaced, each gene must be measured at the same time points.

Differential expression analysis

Computational tools and data normalization. The core differential expression analysis was conducted using the *rpy2* Python package to interface with the R packages *edgeR* and *limma*[38,44]. Genes with low counts were removed using *DGElist* from *edgeR*. Counts were then prepared for differential expression analysis using *voom*[123]. *limma* expects log_2 transformed data, and data that does not conform to this expectation may yield unexpected results. DiffExPy is tailored to analyze time-series data, but the interface with R also enables users to conduct differential expression analysis with static data.

Definition of DiffExPy contrasts. DiffExPy makes contrasts along two dimensions, between conditions and between time points (SI Fig. 3.S1). DiffExPy defines the following classes of contrasts:

(1) **Pairwise (PW)** - calculate LFC at each time point between conditions:

$$\vec{l}_{i,PW} = [\log_2 \frac{\vec{g}^{\,1}_{i|exp}}{\vec{g}^{\,1}_{i|ctrl}}, \ldots, \log_2 \frac{\vec{g}^{\,T}_{i|exp}}{\vec{g}^{\,T}_{i|ctrl}}] \tag{3.9}$$

(2) **Timeseries (TS)** - calculate LFC between each time point and the previous time point for one condition, c:

$$\vec{l}_{i,TS} = [\log_2 \frac{\vec{g}^{\,2}_{i|c}}{\vec{g}^{\,1}_{i|c}}, \ldots, \log_2 \frac{\vec{g}^{\,T}_{i|c}}{\vec{g}^{\,T-1}_{i|c}}] \tag{3.10}$$

(3) **Autoregressive (AR)** - calculate LFC between each time point and the time point before the treatment is applied for one condition, c:

$$\vec{l}_{i,AR} = [\log_2 \frac{\vec{g}^{\,2}_{i|c}}{\vec{g}^{\,1}_{i|c}}, \ldots, \log_2 \frac{\vec{g}^{\,T}_{i|c}}{\vec{g}^{\,1}_{i|c}}] \tag{3.11}$$

Here we assume the treatment is applied at time t=1. The first contrast is equivalent to that of the TS.

(4) **Combinations** - The TS and AR contrasts can also be combined with the PW contrasts to assess if the LFC is significantly different between both the time points and the conditions.

- **PW-TS** - calculate the LFC between conditions and the previous time point:

$$\vec{l}_{i,\text{PW-TS}} = [(\log_2 \frac{\vec{g}^{\,2}_{i|exp}}{\vec{g}^{\,2}_{i|ctrl}} - \log_2 \frac{\vec{g}^{\,1}_{i|exp}}{\vec{g}^{\,1}_{i|ctrl}}), \ldots, (\log_2 \frac{\vec{g}^{\,T}_{i|exp}}{\vec{g}^{\,T}_{i|ctrl}} - \log_2 \frac{\vec{g}^{\,T-1}_{i|exp}}{\vec{g}^{\,T-1}_{i|ctrl}})] \tag{3.12}$$

- **PW-AR** - calculate the LFC between conditions and the time point before the treatment is applied:

$$\vec{l}_{i,\text{PW-AR}} = [(\log_2 \frac{\vec{g}^{\,2}_{i|exp}}{\vec{g}^{\,2}_{i|ctrl}} - \log_2 \frac{\vec{g}^{\,1}_{i|exp}}{\vec{g}^{\,1}_{i|ctrl}}), \ldots, (\log_2 \frac{\vec{g}^{\,T}_{i|exp}}{\vec{g}^{\,T}_{i|ctrl}} - \log_2 \frac{\vec{g}^{\,1}_{i|exp}}{\vec{g}^{\,1}_{i|ctrl}})] \tag{3.13}$$

Because the comparison is to the previous time point, for all comparisons except PW, there are T-1 contrasts. The combination contrasts, PW-TS and PW-AR, provide the most information about when a differential response to a treatment occurs between conditions. However, fewer significant differences are identified because the pooled variances decrease statistical power.

Gene classification

DiffExPy classifies each gene as a differentially expressed gene (DEG), dynamically differentially expressed gene (dDEG), or a differentially responding gene (DRG). Genes are considered DEGs if they pass an F-test described in the *limma* documentation[44]. We enforce that all genes classified as dDEGs and DRGs must also be DEGs. A gene is considered a dDEG if all of its contrasts do not have the same LFC sign, *i.e.* $|\{d \,\forall\, d \in \vec{d}_x\}| > 1$. This filter removes genes that are differentially expressed due to the genetic change but that are not affected by the stimulus. The classification of a DRG is more stringent. A gene is classified as a DRG if the adjusted p-value of the F-test was significant for one of $\vec{d}_{\text{PW-TS}}$ or $\vec{d}_{\text{PW-AR}}$.

There are many ways to assign genes to a cluster depending on the number of time points, treatments, and conditions over which gene expression is measured. Different

methods of clustering test different hypotheses, and we leave flexibility for the user to decide what method is appropriate for their specific application.

Discrete response cluster scores. DiffExPy empirically ranks genes within a discrete cluster (Fig. 3.S11). The cluster score for gene i is computed by calculating the fraction of the sum of the LFC values for each contrast that matches the discrete LFC sign values as follows:

$$CS_i = \frac{\sum\limits_{l \in \vec{l}_x} f(l, p, d)}{\sum\limits_{l \in \vec{l}_x} |l|}, \tag{3.14}$$

where $\vec{l}_x$ is the vector of LFC values for contrast class x, p is the p-value corresponding to each l, and d is the corresponding discrete value of the assigned cluster. The function f weights the calculated LFC by the apparent p value as follows:

$$f(l, p, d) = \begin{cases} |l(1-p)| - |l - l(1-p)| & sgn(l) = d \\ |l - l(1-p)| - |l(l-p)| & sgn(l) \neq d \end{cases} \tag{3.15}$$

Essentially, if a contrast is assigned a nonzero discrete value and the LFC value of the contrast is highly significant (low p-value), most of the LFC value is retained.

CS values are bounded between -1 and 1. Empirically, the cluster score sorts genes by how well they match the discrete path defined by the discrete cluster. It performs poorly for some edge cases, such as when all values in $\vec{d}$ are 0, but these were previously filtered out by the gene classification step.

Model library creation

We generated a library of minimal dynamical systems that was used to simulate time-series expression data that mimics experimental conditions.

Model connectivity. Each model is a directed network with three nodes. Node y represents the output that is matched to measured gene expression. Node G represents the gene that is perturbed, as with a knockout or knockin. Node x is an abstract node that serves two purposes. It summarizes the interactions between G and y with the rest of the genome, allowing for more complex regulatory motifs. Node x is also the target of the externally applied treatment (Fig. 3.S10A). Edges between nodes either activate or inhibit the target node (Fig. 3.S10B).

To limit the regulatory combinations, we prohibited self edges, which yielded 704 unique, weakly-connected networks. We added an input node u to each model to create a controlled forcing function, which represents the external treatment, on node x. The input u was linearly combined with regulation of x by the other nodes.

Model parameterization. We used GeneNetWeaver (GNW) to generate minimal differential equation models for each of the networks and carry out stochastic simulations. GNW models include a protein and mRNA component. Full details are available in the original paper[61]. Systems Biology Markup Language (SBML) style models were generated for each network structure with parameters assigned random values by GNW.

Model parameters were not optimized to improve the fit. It is unclear what objective function—such as minimizing error to LFC or matching expression distributions at each time—would be optimal to fit the parameters. While the parameter space is technically infinite, we relied on GNW to select biologically relevant parameters.

Multiple regulator logic. If a node has two coincident edges, the regulatory logic can be either AND or OR between them (Fig. 3.S10D). However, GNW randomly assigns logic to the regulation of the target mRNA. For each of the 704 network structures, we generated SBML models for each combination of logic, resulting in a library of 2,172

network structure and logic combinations. This library represents a highly constrained structural search space. If the search space were increased to allow self edges, and all combinations of regulatory logic with two or three coincident edges were explored, there would be 102,356 unique models. If the search space included four node networks, an no self edges, there would be 4,870,752 unique networks. We did not simulate these large libraries.

Gene perturbation. For each base model, which we consider the WT model, we created corresponding KO and KI models (Fig. 3.S10C). The modified models have the same parameters as the WT model with minor changes to the mRNA synthesis parameters for node G. In the KO model, the maximum transcription of G was set very low (1e-7), because a value of zero causes integration errors. In the KI model, all upstream regulation of G was removed, and the transcription rate was set to a constant value equal to its maximum rate in the WT model, representing constitutive expression.

Time-series simulations

Stochastic time-series simulations were carried out for each SBML file using GNW. Each stochastic differential equation system was sampled every minute for 1000 minutes. The stimulus was applied to node u at the start and removed halfway through the time series. DiffExPy then sampled the time series at intervals that matched the experimental data. Three independent, stochastic runs of each model were created to represent biological replicates. Microarray-like measurement noise was added to the data, and values were normalized between 0 and 1[61].

Gene Ontology and transcription factor enrichment

An ontology is contructed as a directed acyclic graph (DAG), where the root node has the smallest term depth[21]. In general, more specific GO terms are further down the tree and therefore have higher term depths. Enriched GO terms were called for each class of genes—DEG, dDEG, and DRG—using the same GO DAG, which preserves the depth of the term relationships. We then calculated the term overlap between each combination of exclusive groups. For example the GO terms in the group $DEG \cap dDEG$ are associated with DEGs and dDEGs, but not associated with the DRGs. We calculated p-values to compare the distributions of term depths between groupings using a discrete KS test[141].

To calculate enrichment for associated TFs, gene lists were encoded as TFs using a dictionary of genes associated with each TF. We used a TF association dictionary derived from ENCODE data[127]. TF enrichment was calculated using Fisher's Exact test[49], implemented in *scipy*, comparing the TFs associated with a gene list to the TFs associated with a background gene list. We used different lists of genes as the background depending on the hypothesis being tested. To identify enriched TFs in the set of DRGs, we used the set of all genes as the background. To identify the specific timing of TF enrichment of *SUZ12* and *FOXA1* we used the set of DRGs as the background.

The number of distinct discrete responses increases exponentially with the number of measured time points. For most current experiments, there are few time points and this will not be an issue. However, if more time points were available, specific qualitative behaviors could still be interogated. For example, one could ask "which transcription factors are associated with the set of genes that is initially not differentially expressed before treatment but becomes differentially expressed after X minutes?". The discrete

clusters could easily be used to group genes with this behavior together. Genes A and B may have discrete responses *[0, 0, 0, 1, 1, 1]* and *[0, 0, 0, 0, 1, 1]*, but would be grouped together for enrichment analysis.

Computational development

DiffExPy was developed in Python 3.5.2 using the following major packages: *NumPy and SciPy*[111], *pandas*[142], and *rpy2*. Gene Ontology (GO) enrichment was calculated using the python library *goatools*[143]. The discrete KS test was conducted using the *dgof* package[141]. Figures were generated using *seaborn* and *matplotlib*[115]. The code for DiffExPy can be found on GitHub (*https://github.com/bagherilab/diffexpy*).

3.6.1. Supplementary Information: Additional Figures

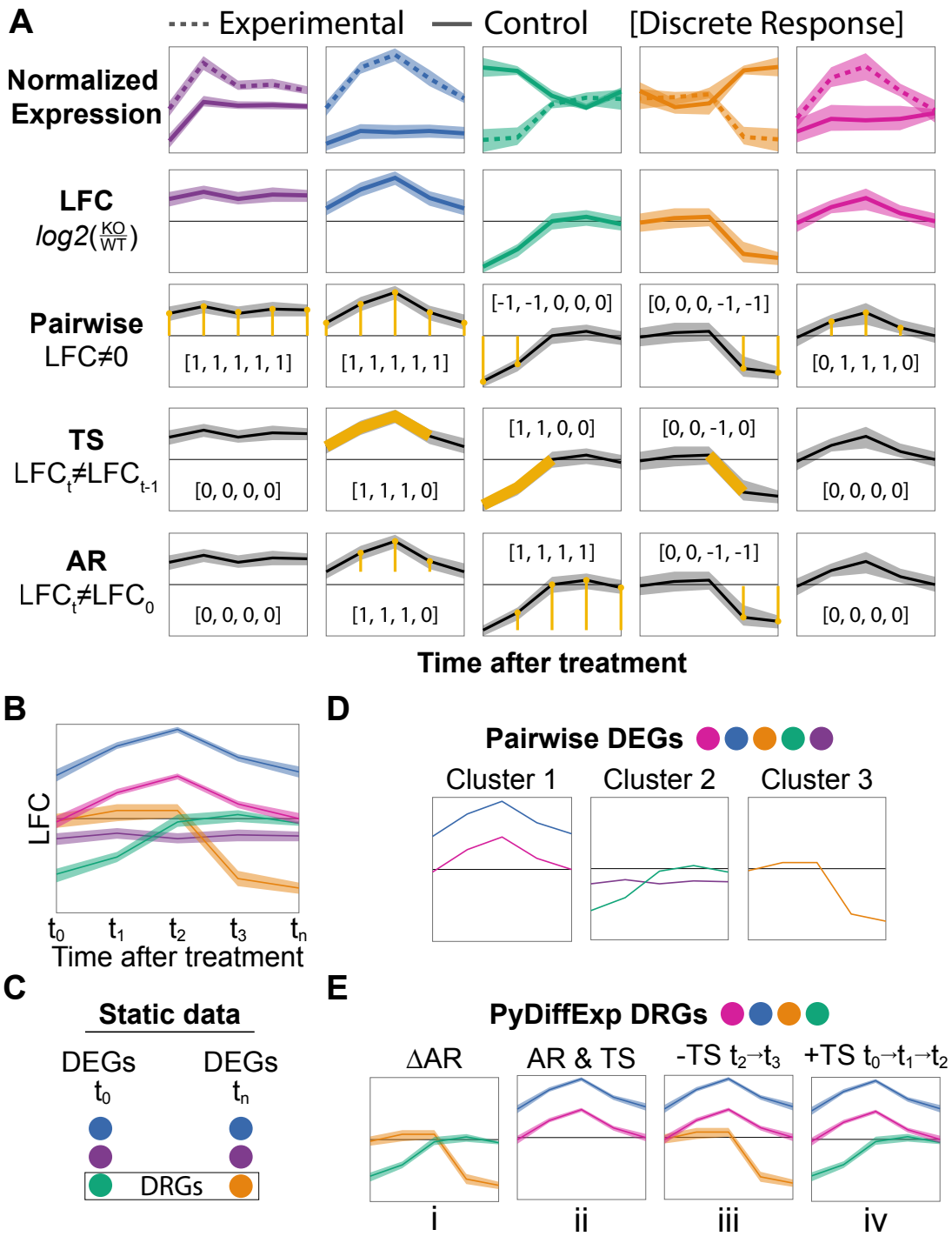


Figure 3.S1. Comparison of strategies to identify and cluster DEGs, dDEGs, and DRGs. (A) Overview of statistical contrasts. Normalized gene expression values, measured for each gene at several time points, were used to calculate the log fold change (LFC) between the experimental (dashed) and control (solid) condition. The lines show the mean expression or LFC, and the shaded area is the error (e.g. 95% confidence interval). Pairwise comparisons independently assess if LFC values are nonzero. Timeseries (TS) comparisons assess if the LFC changed significantly from the previous time point ($l_t \neq l_{t-1}$). Autoregressive (AR) comparisons check if the LFC is different than the initial LFC due to the experimental condition ($l_t \neq l_0$). The discrete responses corresponding to the type of contrast are displayed on the plots. (B) The LFC for example genes measured over time. (C) Sets of DEGs identified by comparing data only at individual time points. DRGs are only identifiable as those whose differential expression at t_0 is statistically different from t_n, which misses many genes. (D) Set of DEGs clustered into groups with similar mean LFC trajectories. DEGs are calculated using an F-test on all pairwise comparisons. Genes can subsequently be clustered by mean LFC using many available clustering methods, such as k-means. (E) Set of DRGs calculated by DiffExPy which includes genes with transient regulation (blue and magenta) that is not captured with only endpoint measurements. Genes that are consistently differentially expressed but show no differential response (e.g. purple) are not considered DRGs. DRGs can be clustered in many ways to find genes that have statistically different expression at later time points (i), have the same response trajectory (ii), or respond similarly at a specific time (iii and iv).

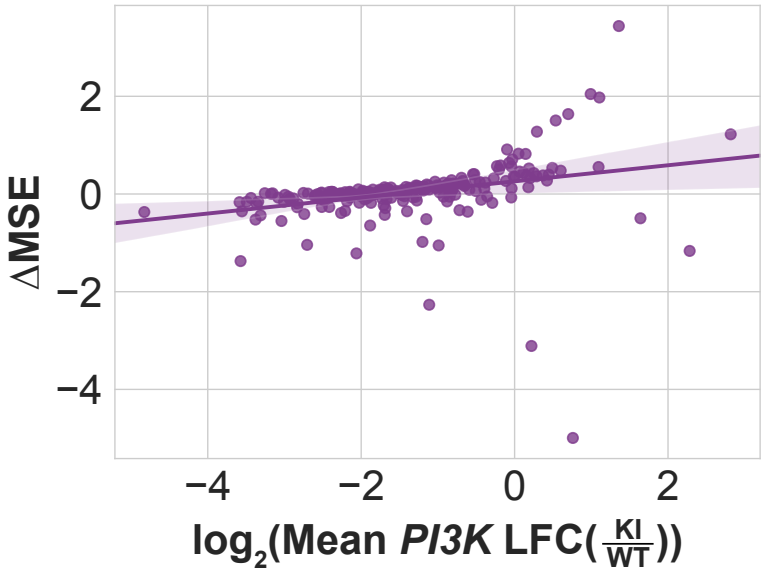


Figure 3.S2. Correlation between the mean LFC between the *PI3K* KI and WT conditions for each gene and the ΔMSE of its trained ensemble model from the null model. Spearman's ρ=0.636, p=5.4e-26

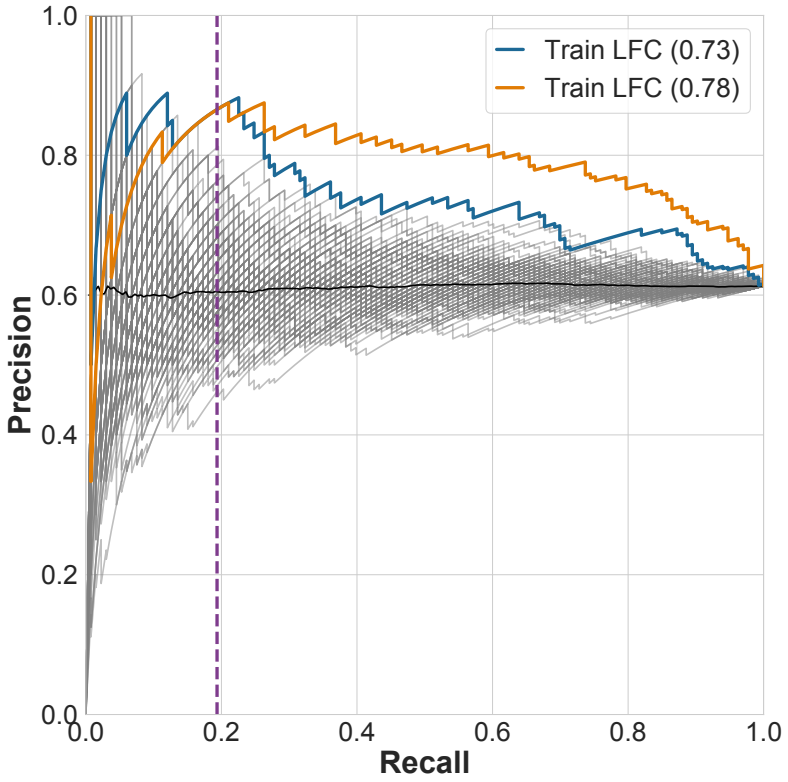


Figure 3.S3. Higher ranked models are more likely to be more predictive than the null model. AUPR curve plot of different methods for sorting gene predictions. Sorting by mean LFC between the training conditions places more accurate predictions at the top of the list. The threshold for selecting more accurate predictions (purple, dashed line) was calculated using the elbow rule of the sorted mean LFC values in the training condition.

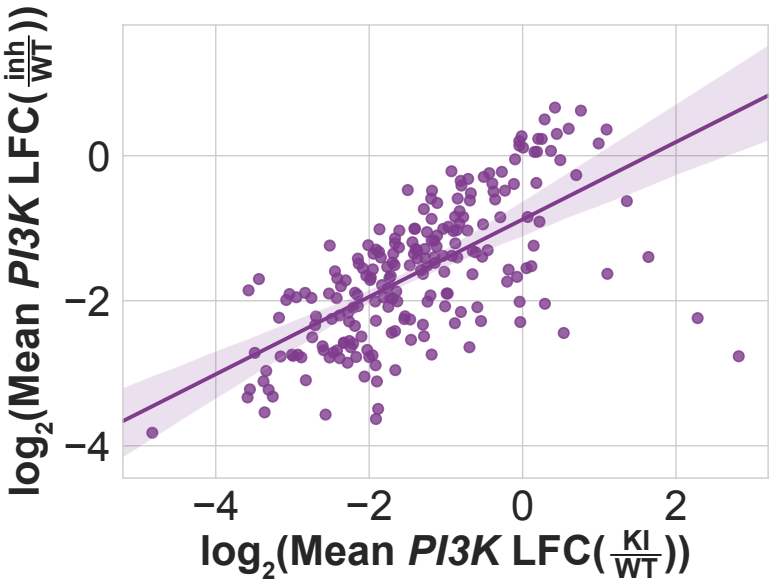


Figure 3.S4. Correlation between the mean LFC between the *PI3K* KI and WT conditions and the mean LFC between the $PI3K^{\text{inh}}$ and WT conditions for each gene. Spearman's ρ=0.684, p=2.7e-31

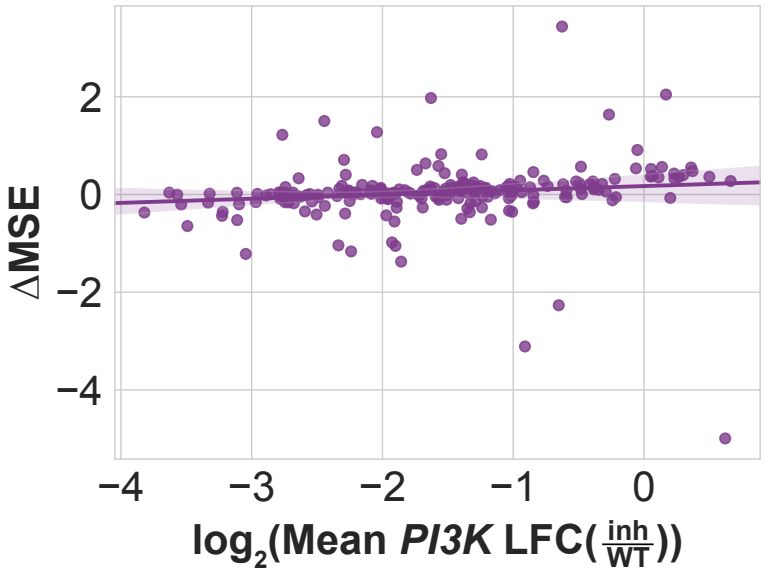


Figure 3.S5. Correlation between the mean LFC between the $PI3K^{\text{inh}}$ and WT conditions for each gene and the ΔMSE of its trained ensemble model from the null model. Spearman's ρ=0.418, p=1.4e-10

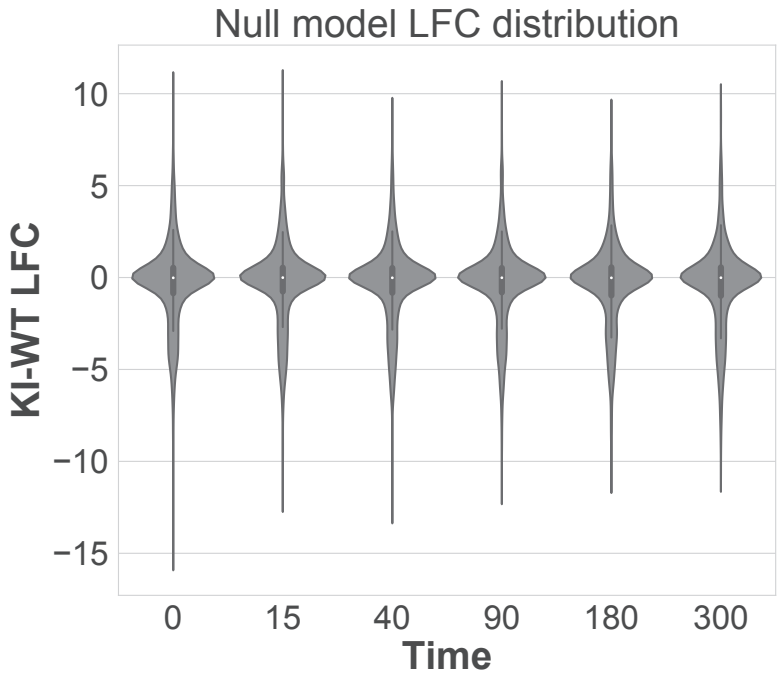


Figure 3.S6. Distributions of predicted LFC values between KI and WT conditions at each time by the simulation library. On average, the library predicts near zero LFC values at each time. However, there is large variation within the library, so the set of models DiffExPy matches to each gene outperform a randomly selected model.

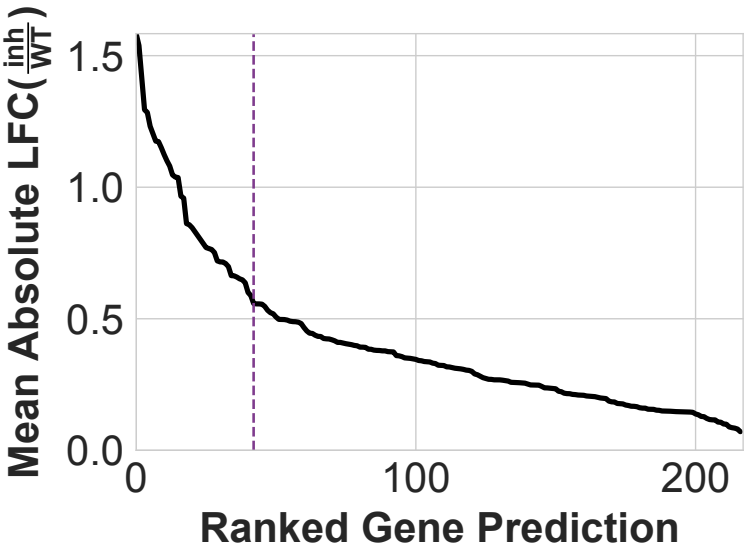


Figure 3.S7. The 217 trained gene models were ranked by the mean absolute LFC of each time point between the $PI3K^{inh}$ and WT conditions. The purple, dashed line shows the cutoff for the top set of genes using the elbow rule.

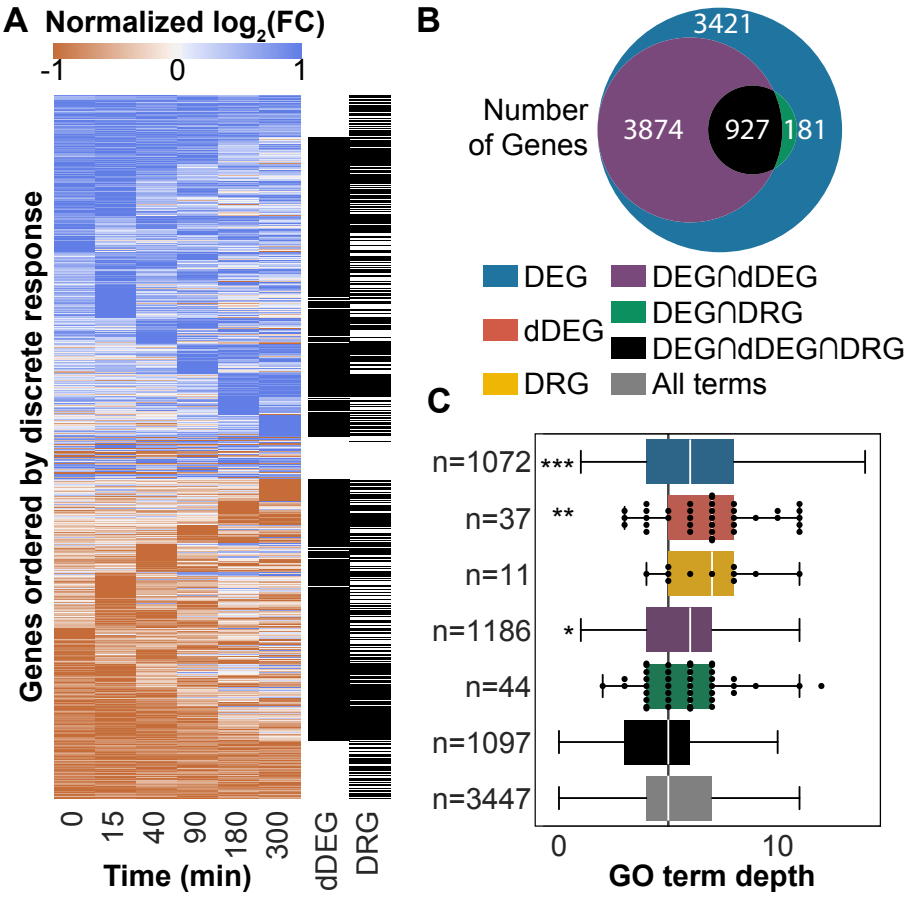


Figure 3.S8. Summary of gene classifications comparing *PI3K* KI (H1047R) to WT. (A) Heatmap of row normalized LFC for all DEGs. Genes that were classified as dDEG or DRG are also labeled. (B) Overlap of genes that were classified as DEG, dDEG, DRG, or some combination. By definition, all dDEGs and DRGs must also be DEGs. (C) Comparison of distributions of GO term depths uniquely associated with intersections of gene sets. p-values were calculated using a discrete KS test. * $p< 0.05$, ** $p< 0.01$, *** $p< 0.001$.

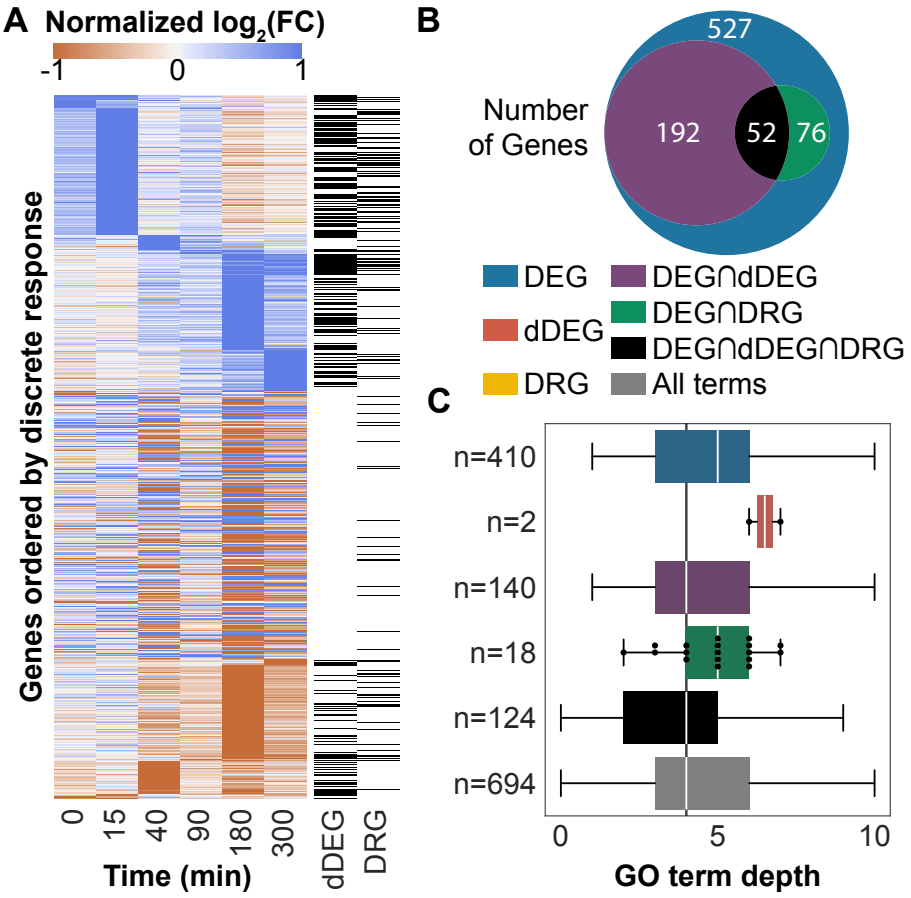


Figure 3.S9. Summary of gene classifications comparing *PI3K*$^{\text{inh}}$ (A66 treatment) to WT. Few genes were significantly impacted by the *PI3K*$^{\text{inh}}$, which limits how many GO terms can be found. (A) Heatmap of row normalized LFC for all DEGs. Genes that were classified as dDEG or DRG are also labeled. (B) Overlap of genes that were classified as DEG, dDEG, DRG, or some combination. By definition, all dDEGs and DRGs must also be DEGs. (C) Comparison of distributions of GO term depths uniquely associated with intersections of gene sets.

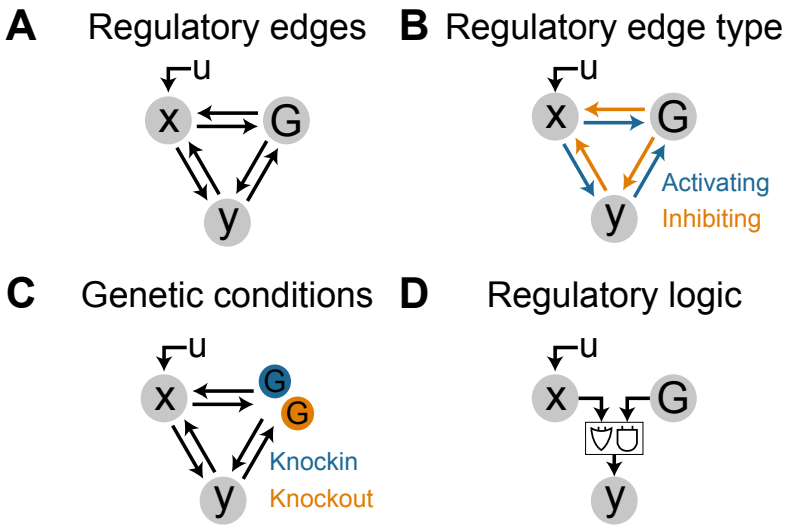


Figure 3.S10. In silico simulation library model constraints. (A) All unique combinations of three-node networks without self-edges were generated. Node y is the output node whose simulated expression is matched to discrete trajectories of genes in the experimental data. Node G is the experimentally perturbed node. Node x represents interactions with the rest of the genome. A forcing function, u, is applied to node x to simulate the application of a treatment. (B) Edges between nodes can be activating or inhibiting. (C) Simulations were conducted where node G is knocked out (no expression in the simulation) or knocked in (constituitively expressed in the simulation). (D) If there are two upstream regulators of a node, combinations of the regulators interacting with AND or OR logic were created.

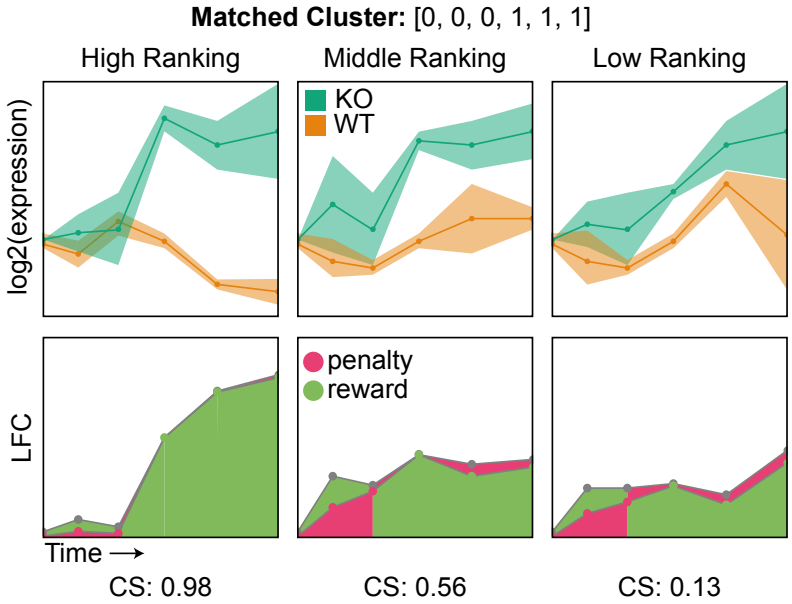


Figure 3.S11. Cluster score of genes or model simulations. The high ranking gene (left) has a cluster score (CS) close to 1 because it has a low LFC at the early time points and highly significant LFC at later time points which match the assigned cluster. The low ranking gene (right) still qualitatively matches the assigne cluster. However, it has a lower CS because the LFC values at early time points are higher and the LFC values at later time points are not as significant.

CHAPTER 4

Gene expression dynamics reveal Sprouty mediated cross-talk of Wnt pathway genes

This work is in preparation to be submitted with Behnam Nabet, Jon Licht, and Neda Bagheri to PLOS Computational biology. A modified version of it may be published after peer review.

Abstract

Sprouty genes (*Spry*) are feedback regulators of receptor tyrosine kinase signaling, and are genes with known tumor suppressing activity. The impact of *Spry* induction is ligand specific, but it is well-understood to be a negative feedback regulator of the mitogen-activated protein kinase pathway in response to fibroblast growth factor (FGF). However, the precise impact *Spry* has on transcription resulting in tumor suppression remains unknown. We use DiffExPy to analyze time-series gene expression data in $Spry124^{fl/fl}$ and $Spry124^{-/-}$ murine embryonic fibroblasts (MEFs). We show that the expression dynamics of *Spry* in response to FGF encode transcriptional regulation that spans RNA production, metabolism, and apoptotic processes. We also present results that suggest *Spry* regulates the angio- and oncogenic gene connective tissue growth factor (*Ctgf*) via the Wnt signaling pathway. Our results clarify the transcriptional role of *Spry*, and provide quantitative, testable models of gene expression that can be applied and validated in different experimental contexts.

Author summary

A major goal of systems and computational biology is to disentangle the complex signaling pathways that govern cellular behavior. In each cell, a few core pathways are responsible for integrating multiple external stimuli and determining the cell's response. Sprouty proteins provide feedback regulation of the central mitogen activated protein kinase (MAPK) pathway. Prior work demonstrated that Sprouty loss is not synonymous with mutations to MAPK components that are associated with cancer. To understand how Sprouty regulates expression we measured gene expression in mouse embryonic fibroblasts at several time points after treatment with fibroblast growth factor. We use a top-down analysis of the gene expression dynamics to propose Sprouty mediated crosstalk between the MAPK and Wnt pathways that regulates the expression of angiogenic factors. Our results provide a new regulatory role for Sprouty in determining cellular fate in response to stimuli. More generally our work demonstrates how time series data can be used to decode regulatory processes.

Introduction

Sprouty (*Spry*) was originally discovered in *D. melanogaster* as a common antagonist of the FGF signaling pathway[144,145]. Four mammalian homologs were since identified and shown to play a critical role in development[146,147]. *Spry* genes are feedback regulators of receptor tyrosine kinase (RTK) signaling that have been shown to modulate the activity of Ras proteins and the rest of the downstream mitogen-activated protein kinase (MAPK) pathway[147]. The exact mechanism of *Spry* regulation remains unknown, but data suggest that *Spry* proteins sequester targets to plasma membranes by binding phosphatidylinositol-4,5-bisphosphate[147,148].

The role of *Spry* genes during development has been extensively studied, and they were shown to regulate angiogenesis in mammalian endothelial cells[145,147,149,150]. Down-regulation of *Spry* is observed in many cancers, and and *Spry* proteins exhibit tumor suppressing activity via negative feedback control of Ras targets[151–153]. Previous work also revealed that *Spry* loss uniquely alters the gene expression and enhancer landscape of murine embryonic fibroblasts (MEFs) compared to *HRas* mutation, suggesting a more complex role for *Spry* in transcriptional regulation[154]. Understanding the transcriptional role of *Spry* is complicated by interactions with other pathways such as Wnt and NF-κB[153,155,156], however with time-resolved gene expression data, it is easier to decode the effect of *Spry* on the transcriptional landscape and ultimately tumorigenesis.

We collected time-series gene expression data from genetically matched *Spry124*$^{fl/fl}$ (WT) and *Spry124*$^{-/-}$ (KO) MEFs after treatment with fibroblast growth factor (FGF). We used DiffExPy this time-series gene expression data[121]. Our analysis yields insights into transcriptional regulation at several genomic scales. By applying DiffExPy to time-series gene expression data, we identify both global and gene-specific differential responses to FGF caused by *Spry* loss. We then train quantitative models that match measure gene expression dynamics and generate testable hypotheses of transcription regulation. Our work provides a framework for future studies to develop integrated experimental and computational pipelines that create quantitative models of gene regulation that can be readily adapted to other systems.

Results

We measured gene expression in WT and KO MEFs at 0, 15, 60, 120, and 240 minutes after FGF treatment (Fig. 4.1A). To gain global understanding of *Spry* regulation in response to FGF, we analyze the time-series data with DiffExPy. Our analysis provides

three key insights into *Spry* regulation. First, we use the time-series data to decouple the genetic effect of *Spry* knockout on gene expression from the role *Spry* plays in the transcriptional response to FGF. Second, we associate the discrete response clusters generated by DiffExPy with enriched Gene Ontology (GO) terms and transcription factors (TFs), revealing differential responses that might RNA production, metabolic, and apoptotic processes. Lastly, we demonstrate how the gene expression dynamics suggest that *Spry* regulates the expression of the oncogene connective tissue growth factor (*Ctgf*) via the Wnt pathway, in addition to the canonical MAPK pathway.

Time-series measurements reveal varied differential expression responses to FGF treatment

To demonstrate the importance of time-series measurements of gene expression for understanding cellular regulation, we highlight results from five categories of genes (Fig. 4.1B). The reference genes *Fos* and early growth response 1 (*Egr1*) behave as expected in response to FGF; their expression peaks 60min after FGF treatment, and is unaffected by *Spry* KO. Previously identified responders to epidermal growth factor—such as the dual-specificity phosphatases (*Dusp*s) that act as feedback attenuators, and zinc finger protein 36 (*Zpf36*) and kruppel-like factor 2 (*Klf2*) that act as early expression regulators[157]—also respond to FGF treatment, but are differently impacted by *Spry* KO. As expected, the *Spry* target genes semaphorin 7A (*Sema7a*) and serpin family B member 2 (*Serpinb2*) exhibit strong differential responses to FGF treatment between the WT and KO conditions[154].

Lastly, to emphasize the importance of properly analyzing time-series measurements of gene expression, we show responses for pleiotrophin (*Ptn*) and Wnt family member 7B

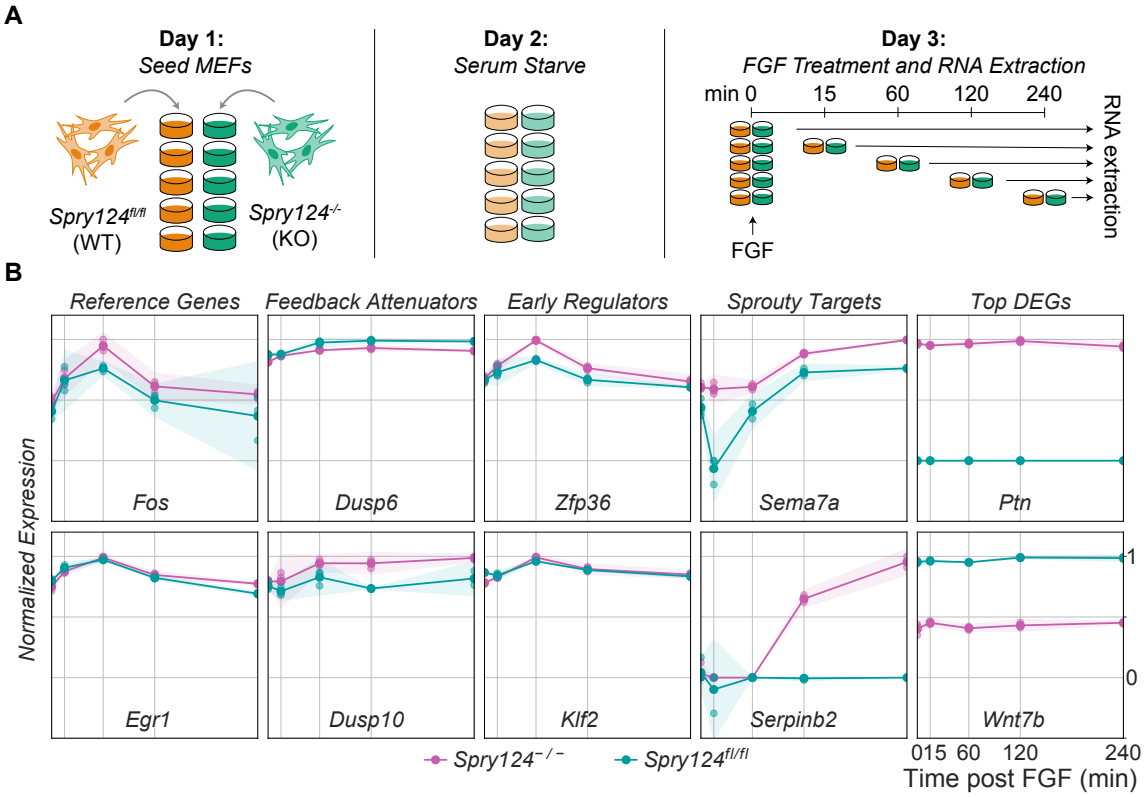


Figure 4.1. Overview of experiments and reference genes. (A) Experimental procedure of RNA quantification in MEFs. *Spry124* WT and genetically matched KO cells were plated in growth media on day 1. MEFs were then serum starved for 24hrs. On day 3 MEFs were treated with FGF for 0, 15, 60, 120, and 240 minutes before total RNA extraction was performed. Four biologically independent samples were measured for each condition at 0 minutes, and three for each other time point.(B) Example genes from five categories highlight the importance of time-series gene expression data. Gene expression values were normalized to their maximum expression value for display. The shaded region shows the 95% confidence interval of the mean expression value for easier comparison of significant differences.

(*Wnt7b*) (Fig. 4.1B). A typical differential expression analysis would likely rank both of these genes as significantly differentially expressed. However, the time-series profiles of these genes illustrate that their differential expression is primarily driven by the *Spry* KO, and they are unaffected by FGF treatment.

Gene expression captures variation in condition and time

We conducted principal component analysis (PCA) on the normalized gene expression data to identify natural variation in the data. Principal component 1 (PC1) separates samples by genetic condition and PC2 separates samples mostly by the time the sample was take after FGF treatment (Fig. 4.S1A). Combined, PC1 and PC2 capture almost 50% of the variance (Fig. 4.S2). These PCA results indicate that there is an observable difference in gene expression along the two dimensions that were tested, *Spry* condition and time after FGF treatment. They also confirm that there is no systematic bias in the data samples that needs to be corrected.

Sprouty knockout enriches GO terms and TFs

Differential gene expression analysis of the data at baseline, prior to FGF treatment indicates that the *Spry* knockout significantly alters the baseline expression of many genes (Fig. 4.S1B). To deconvolute the changes in gene expression caused by the *Spry* KO from those caused by FGF treatment we used DiffExPy to classify genes as differentially expressed genes (DEGs), dynamic DEGs (dDEGs), and differentially responding genes (DRGs) based on their discrete responses[121]. Overall, we classify 6,835 genes as DEGs. Of these, we identify 3,631 dDEGs, 612 DRGs, and 445 genes that are classified as both (Fig. 4.2A). Each gene class is enriched for association with many GO terms (Fig. 4.2B). The terms associated with DEGs, dDEGs, and DRGs generally have lower fold enrichment, but more significant p-values, in corresponding order. This result is likely due to the number of genes in each category. The GO terms with the highest fold enrichment associated with DEGs, dDEGs, and DRGs are generally related to RNA processing, cell growth, and cell differentiation, respectively.

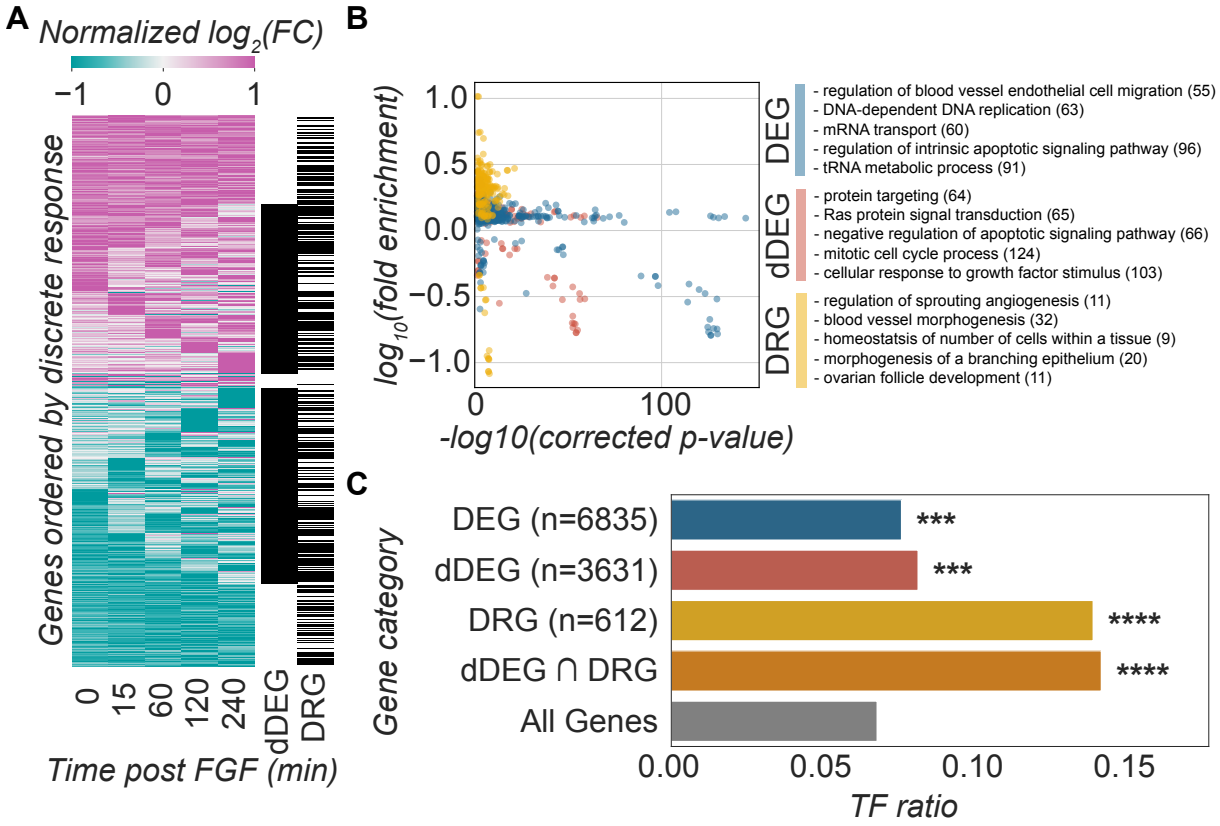


Figure 4.2. Gene classification and TF enrichment. (A) Heatmap of row normalized LFC values for all DEGs. Genes that are classified as dDEGs or DRGs are also labeled. (B) Scatter plot of enriched GO terms associated with gene sets for each classification. The top five unique GO terms with the highest fold enrichment are shown for each classification. (C)TFs are significantly over-represented in all categories of genes. p-values were calculated using Fisher's exact test. *** $p < 0.001$, **** $p < 0.0001$.

We also find that TFs are significantly over represented in the set of genes classified as DEGs, dDEGs, DRGs and the intersection of dDEGs and DRGs (Fig. 4.2C). These results suggest the transcriptional response to FGF treatment is significantly altered when *Spry* is knocked out. The set of DRGs has nearly double the fraction of TFs as the set of all genes. This result highlights how dynamic changes in transcription factor expression is critical to the cellular response to a stimulus. We next analyze these genes in depth to better understand *Spry*'s role in transcription.

We also created a correlation network of the DRGs that are TFs to further understand their regulatory roles in response to FGF Fig 4.S5. The network contains submodules of highly correlated TFs that recapitulate known connections. The *Mapk* genes *Egr2*, *Junb*, and *Klf2*, and members of the minichromosome complex *Mcm4*, *Mcm5*, and *Mcm6* are connected in separate modules. Other oncogenes, such as homeobox A2 (*Hoxa2*), A5 (*Hoxa5*) and twist family member 2 (*Twist2*) are represented in the network and were previously shown to respond to *Spry* KO[154].

Gene expression dynamics reveal regulatory programs

Applying DiffExPy to time-series data we can learn about the precise timing of regulatory events governing transcription. Using the discrete pairwise clusters for the set of all dDEGs and DRGs we performed a search of the enriched GO terms and TFs associated with each growing cluster (Fig. 4.S3). At each time point, the discrete LFC can be positive, negative, or zero. As more time points, T, are added, the number of theoretical clusters is therefore 3^T.

Overall, the function of the enriched GO terms can be categorized by the initial effect of the *Spry* knockout. Genes that are initially overexpressed are associated with protein modification and cell survival terms, which is in line with *Spry*'s known function. Put differently, *Spry* is known to inhibit the MAPK pathway, which in turn governs key cell growth and survival genes; thus, when *Spry* is not expressed, genes related to cell growth and survival are expected to become overexpressed. Genes that are initially underexpressed are primarily associated with terms related to RNA processing, while genes that are not significantly differentially expressed are associated with terms related to mitochondria and metabolism (Fig. 4.S3).

The GO enrichment trends also align with the TF enrichment trends. For example, the TFs associated with initially overexpressed genes include *ELK* proteins, and CCAAT/enhancer-binding proteins (*CEBP*s), which are known regulators within the MAPK pathway[158]. Additionally, TFs associated with genes that are not differentially expressed before FGF treatment include forkhead box proteins O1 and O4 (*Foxo1 and Foxo4*). These TFs are known to be in the glugacon signaling and insulin resistance pathways, suggesting a possible connection to metabolism[158].

Differentially responding genes impact angiogenesis

The tumor suppressing activity of *Spry* is typically associated with its function as a negative regulator of FGF and other RTK signaling[147]. Our results confirm that *Spry* loss is associated with overactive *Ras* mutations, and the observed trends in GO enrichment (Fig. 4.S3) also support of those phenotypic changes[159]. However, analyzing the set of DRGs provides unique insight into *Spry* regulation, which suggests that *Spry* may mediate cross-talk between several central pathways. By definition, the set of DRGs are the genes with the most statistically significant changes in the time-series data. These genes are associated with many GO terms, including 11 associated with *regulation of sprouting angiogenesis* and 32 associated *blood vessel morphogenesis* (Fig. 4.2). The DRGs associated with these enriched terms include *Angpt1, Vegfa, Vegfb, Junb, Pik3r2, and Elk3*. This result is consistent with previous literature. Sprouty proteins are known to impact branching events, including angiogenisis, likely through their regulation of vascular endothelial growth factor (VEGF)[147,160].

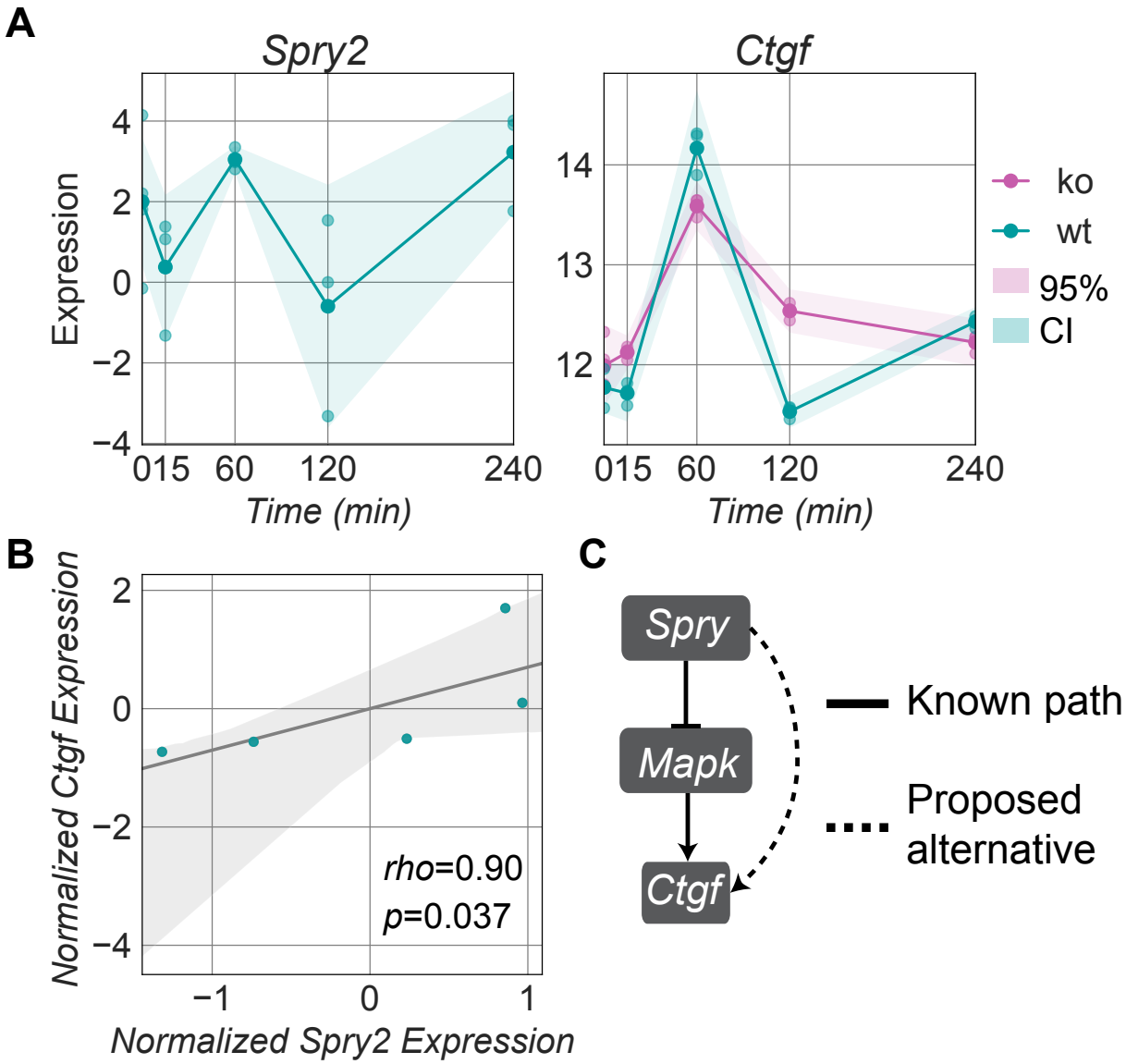


Figure 4.3. Observed *Ctgf* expression does not match known pathway logic. (A) Time-series plot of *limma* normalized intensity values of *Spry* and *Ctgf*. The *Spry* KO data is not shown in the *Spry* plot as it is below the noise threshold (B) Joint distribution of WT expression values between *Spry* and *Ctgf* (C) Known pathway between *Spry* and *Ctgf* with an alternate proposed pathway that exhibits positive regulation.

Spry cross-talk with the Wnt pathway regulates angiogenic factors

Included in the DRGs associated with angiogenesis are *Hes1, Ctgf, and Tead2* and several other genes in the Wnt pathway[158]. Previous research demonstrated important cross-talk

between *Spry* and the Wnt pathway during development[155,161,162]. Here we use the additional temporal information encoded in the gene expression dynamics of Wnt pathway genes to identify specific changes in proto-oncogenes typically associated with angiogenesis. We propose *Spry* mediates cross-talk between the MAPK and Wnt pathways in an oncogenic context.

Ctgf is a secreted protein involved in angiogenesis, and a known oncogene[163,164]. It is also known to be regulated by the MAPK pathway[158,163]. We observed that *Ctgf* expression was not initially differentially expressed between the WT and KO conditions, but showed a significant differential response to the FGF treatment (Fig. 4.3A), indicating that *Ctgf*'s expression is regulated in part by *Spry*. However, we find that *Spry2* wild-type expression is positively correlated with *Ctgf* (ρ=0.59, p=0.017, Fig. 4.3B). This correlation is opposite of what we would expect if *Ctgf* is regulated by the MAPK pathway (Fig. 4.3C)[165].

We sought to explain this discrepancy by analyzing the other angiogenic DRGs. We find correlations between several angiogenic DRGs in the Wnt pathway and *Ctgf* (Figs. 4.4A and 4.S4). Based on these results, we suggest a model with an inhibitory cascade between *Spry* and *Ctgf* (Fig. 4.4C). Our model proposes three novel interactions within the known Wnt pathway (Fig. 4.4C). First, *Spry* acts as an activator of Wnt by preventing the inhibition of Wnt by FGF, which was previously shown to occur in a developmental process[161]. Second, we observe a strong negative correlation between *Hes1* and *Tle1* (Fig. 4.S4), which are known to interact[30]. Finally, we propose negative regulation of *Ctgf* by *Tle1*, which are also negatively correlated (Fig. 4.S4). This model is consistent with our observations and the current understanding of Wnt signaling, but must still be validated experimentally[30,158,166].

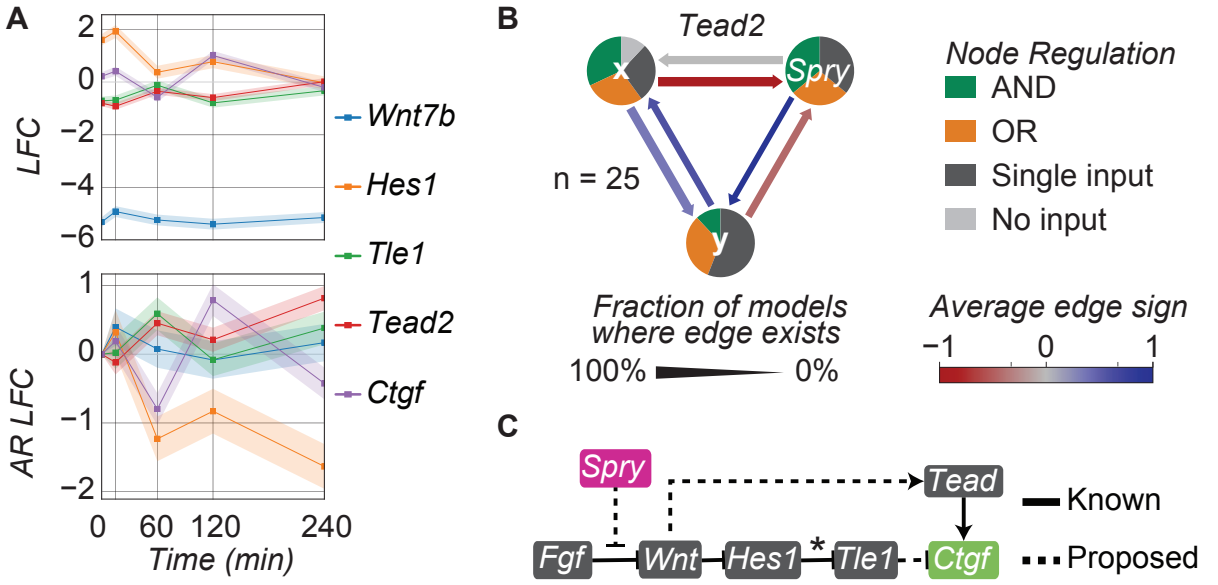


Figure 4.4. *Spry* regulates *Ctgf* via the Wnt pathway. (A) Time series of LFC values and autoregressive (AR) LFC values of the pathway genes over time. The comparison to the zero time point removes the genetic effect of the *Spry* knock out and highlights the effect of the FGF stimulus. (C) Proposed model of regulation. Known interactions are shown as solid lines, and proposed interactions as dashed lines. The interaction between *Hes1* and *Tle1* is starred because it is a known interaction, but our analysis indicates an inhibitory interaction.

DiffExpy suggests models in which *Spry* regulates *Tead2* expression

Using DiffExPy we trained ensemble models to the discrete expression profiles of all the dDEGS. We highlight the models matched to *Tead2* because it is a known regulator of *Ctgf* and is one of the DRGs associated with angiogenesis[167]. Our previous work also revealed that *Tead2* is overrepresented in *Spry* WT superenhancer regions[154]. The models matched to *Tead2* reveal possible modes of regulation of these genes by *Spry* (Fig. 4.4B). 12 of the 25 networks include direct activation between *Spry* and *Tead2*. All 11 of the networks with an indirect connection from *Spry* to *Tead2* through node x have activating logic.

This large fraction of trained models with positive regulation from *Spry* to *Tead2* suggests that the observed gene expression dynamics of *Tead2* are most likely explained

by activation via *Spry*. The activation of *Tead2* could in turn contribute to the observed activation of *Ctgf*. Previous research suggests cross-talk between the Wnt and *Hippo* pathways[168,169], which could also mediate the observed change in *Ctgf* expression (Fig. 4.4C).

Interestingly, *Tead2* also has several interesting properties within the transcription factor network (Fig. 4.S5). It is more connected than many of the TFs in the network with a degree of 6 as compared to the median of 3. The path lengths between *Tead2* and the Wnt genes *Tle6*, *Hes1*, and *Tcf3* are just 2, while the average path length in the network is 4.7. These properties suggest that *Tead2* provides an important bridge between regulatory modules that govern the response to FGF.

Discussion

While Ras is frequently mutated in cancer, it is a challenging target to directly inhibit[7,170]. If *Spry* facilitates tumor suppression by regulating other pathways, it may provide an easier route for therapeutic development.

Ras mutations are common in a variety of cancers, but the complexity of the MAPK pathway and difficulty targeting Ras hampers the development of effective therapies[170]. Sprouty proteins exert negative feedback regulation of Ras and other RTK signaling molecules, but their precise role in the transcriptional landscape is unknown[147]. Previous work demonstrated that Sprouty loss produces transcriptional effects distinct from Ras mutations, suggesting a more complex role for Sprouty[154]. Identifying Sprouty's dynamic regulatory role of transcription in tumorigenesis will reveal therapeutic targets.

We used time-series gene expression data from *Spry124*$^{fl/fl}$ and *Spry124*$^{-/-}$ MEF populations stimulated with FGF to identify *Spry*'s role in regulating transcription. We applied

DiffExPy to analyze the gene expression dynamics and uncover regulatory insight at several genomic scales. On the global scale, we identified key biological processes changed by the Sprouty knockout. As an example, we demonstrate a specific analysis of angiogenic factors that suggest regulation of *Ctgf* inconsistent with its known regulation. We proposed regulation of Wnt pathway genes, including *Ctgf*, by *Spry* (Fig. 4.4). Previous studies suggest interactions between FGF and Wnt in a developmental context, but we highlight relevant cross-talk between the two pathways in an oncogenic context[155,162,171].

We also used the discrete response clusters generated by DiffExPy to identify trends in GO term and transcription factor enrichment over time (Fig. 4.S3). These results suggest three distinct programs that differentially respond to FGF treatment after *Spry* is knocked out. Initially overexpressed genes are primarily enriched for GO terms related to cell survival and TFs in the central MAPK pathway, even as they exhibit varied responses to FGF. This pattern is consistent with *Spry*'s role as a negative regulator of the MAPK pathway. We also observed enriched terms related to RNA regulation and metabolism for genes that were initially underexpressed and those that were not differentially expressed, respectively. These differences may be necessary to support the phenotypic changes induced by FGF treatment or may indicate additional misregulation in the absences of *Spry*.

The results from our analysis using DiffExPy suggest regulation of many genes by *Spry* in response to FGF. Many of the dDEGs were matched to ensemble models of minimal SDE systems. We highlight the regulatory insights proposed by these models using *Tead2* as an example. Our results provide testable hypotheses that must be experimentally validated. However, unlike typical gene expression analyses, the models trained by DiffExPy offer clear, quantitative predictions that can be corroborated. Much of the initial

regulation in the MAPK and Wnt pathways may not be driven by changes in expression. Therefore, integration and validation with proteomic and other genomics tools may clarify any discrepancies. Additionally, quantifying the phenotypic effects of *Spry* knockout and FGF treatment would help anchor the desired output of future mathematical models[51].

In this study, we manually interrogated the genes related to angiogenesis. However, there are many clusters of genes that are enriched for different GO terms and TFs. DiffExPy could be extended to conduct gene set enrichment analysis[22] on these clusters to identify pathways that are also enriched. It could also be integrated with other services such as KEGG[158], Biogrid[28,30], and StringDB[31] to programmatically find plausible paths of regulation from prior knowledge, and provide new hypothetical interactions.

Overall, we used time-series data to provide insight at several genomic scales. We propose predictive models for many genes that could not be generated using traditional differential expression analyses without many, diverse treatments. There are many more pathways that could be explored in this dataset by automating prior knowledge pathway searches. Our approach can be readily applied to gain insight in other systems for which time-series data exists, or is gathered.

Materials and Methods

Cell lines and microarray profiling

Spry124$^{fl/fl}$ and *Spry124*$^{-/-}$ murine embryonic fibroblasts (MEFs) were cultured in DMEM containing 10% FBS, 100 U/mL penicillin, and 100 μg/mL streptomycin at 37 ºC and 5% CO2, as previously described[154]. For microarray profiling studies, biologically independent cultures of MEFs were plated in 10 cm plates and allowed to adhere overnight. MEFs were serum starved in starvation media (DMEM containing 0.2% bovine serum albumin,

100 U/mL penicillin, and 100 μg/mL streptomycin) for 24 hours. Prior to treatments, starvation media was removed, and fresh starvation media was added to all plates. After equilibration, MEFs were treated with fibroblast growth factor (FGF, Thermo Fisher Scientific) as indicated, lysed, homogenized using Qiashredder columns (Qiagen), and RNA was extracted using the RNeasy Plus Mini Kit (Qiagen), according to manufacturer's instructions. Samples were collected 0, 15, 60, 120, and 240 minutes after FGF treatment (Fig. 4.1A). Biologically independent samples were collected for for each treatment and condition (n=4 for 0 minute samples and n=3 for all other time points). RNA was hybridized to MouseRef-8 v2.0 Expression BeadChip (Illumina, BD-202-0202) by the Genomics Core at the Cleveland Clinic Foundation Lerner Research Institute. Gene level expression values were calculated as the mean of all probe values corresponding to each gene.

RNA quantification and normalization

Microarray intensity values were quantified by the Genomics Core at the Cleveland Clinic Foundation Lerner Research institute using Genome Studio (v. 1.9.0). Probe level readings were corrected by background subtraction. Gene level quantification was calculated as the mean of all probe level values for each gene.

Additional normalization was conducted using the default functionality provided by DiffExPy. Gene expression intensity values were log_2 transformed. Log fold change (LFC) values were calculated from the transformed intensity values using linear modeling and subsequent empirical bayes estimation by *limma*[44].

DiffExPy parameters

DiffExPy was run with default parameters. All p-value thresholds used, including adjusted p-values for GO and TF enrichment, were 0.05. Ensemble models matched to discrete responses of experimental gene expression were required to have a cluster score and mean correlation greater than zero. A complete description of the DiffExPy parameters is available in the original manuscript. DiffExPy defines three categories of discrete gene response behaviors. The categories are differentially expressed genes (DEGs), dynamic DEGs (dDEGs), and differentially responding genes (DRGs). Gene Ontology (GO) enrichment for these categories was conducted using PANTHER[23].

Computational development

All analysis was conducted using Python 3.5.2 using the following major packages: *NumPy and SciPy*[111], *pandas*[112], and *scikit-learn*[114]. Figures were generated using *seaborn* and *matplotlib*[115]. The GO term enrichment search of the gene clusters was conducted using *goatools*[143]. Other enrichment tests used Fisher's exact test[49] implemented in *SciPy*.

Acknowledgements

This research was supported, in part, by a Nicholson Fellowship and Biotechnology Training Program Fellowship (to J.D.F), NSF CAREER Award (CBET-1653315 to N.B.), and the McCormick School of Engineering at Northwestern University.

4.1. Supplementary Information

Principal component analysis. Principal component analysis (PCA) was conducted using *sklearn*. The log_2 expression values were mean-centered and scaled using the *zscore* for each gene.

Transcription factor network inference. A transcription factor regulatory network was inferred using the set of 85 genes that are both DRGs and identified as mouse transcription factors (DRG-TFs) in the Riken Transcription Factor Database[172]. There were too few time points to make use of network inference algorithms tailored to using time-series data[48]. Therefore, to identify DRG-TFs with similar expression profiles over time and across conditions we first created feature vectors for each DRG-TF. Each vector is a concatenation of the independently *z-scored limma* coefficients from the default DiffExPy contrasts. Overall, the feature vectors capture how the DRG-TFs respond to the FGF treatment in each condition, between conditions, and across time.

Spearman's rank correlation was calculated between all pairwise combinations of DRG-TF vectors. Adjusted p-values were calculated using the Benjamini-Hochberg procedure with a false discovery rate (FDR) of 0.0005[173]. A network of 76 nodes and 128 edges was created at this FDR (Fig. 4.S5). The network was visualized with Gephi 0.9.2[174]. The modules were computed using the default community detection algorithm in Gephi with a resolution of 1[175].

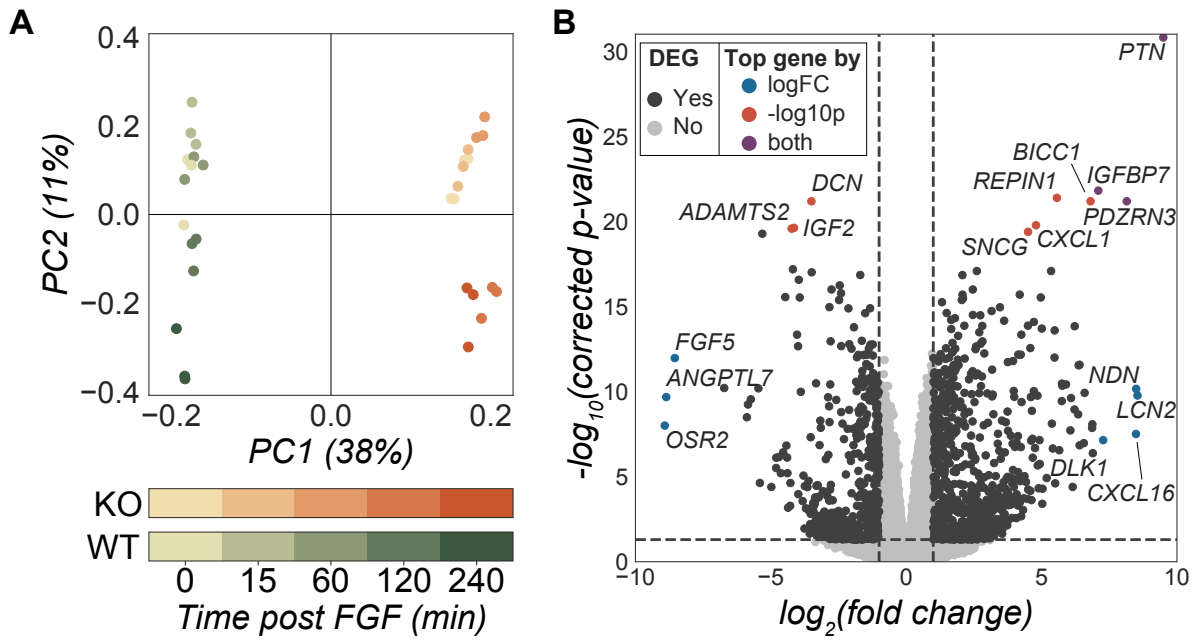


Figure 4.S1. Deconvolution of the genetic and FGF treatment effects on *Spry* expression data. (A) Loadings plot from the Principal Component Analysis of *Spry* data. Points show the weight of samples in PC1 and PC2 space. PC1 explains 38% of the variance in the data and separates samples by genetic condition. PC2 explains 11% of the variance in the data and mostly separates samples by the time the sample was measured after FGF treatment. (B) Volcano plot of log fold change in expression between *Spry* WT and KO samples at baseline, prior to FGF treatment. The top 15 genes with the most significant differential expression at this time point are highlighted in green. Other significant differentially expressed genes (based on the thresholds in orange) are shown in purple. An adjusted p-value of 0.05 and an absolute $log2$ fold change of 1 were used as thresholds for significance.

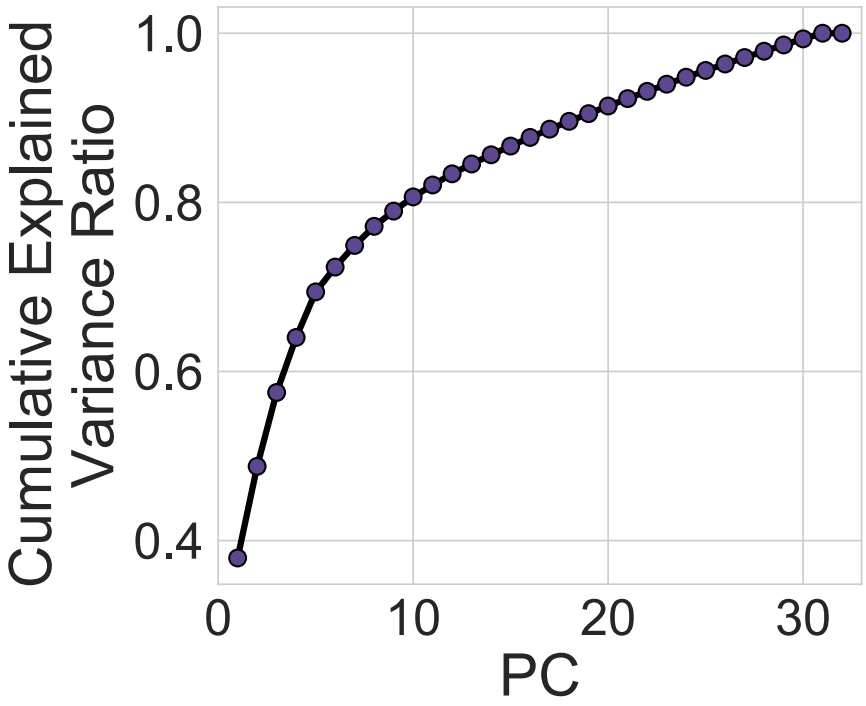


Figure 4.S2. Explained variance of PCA decomposition

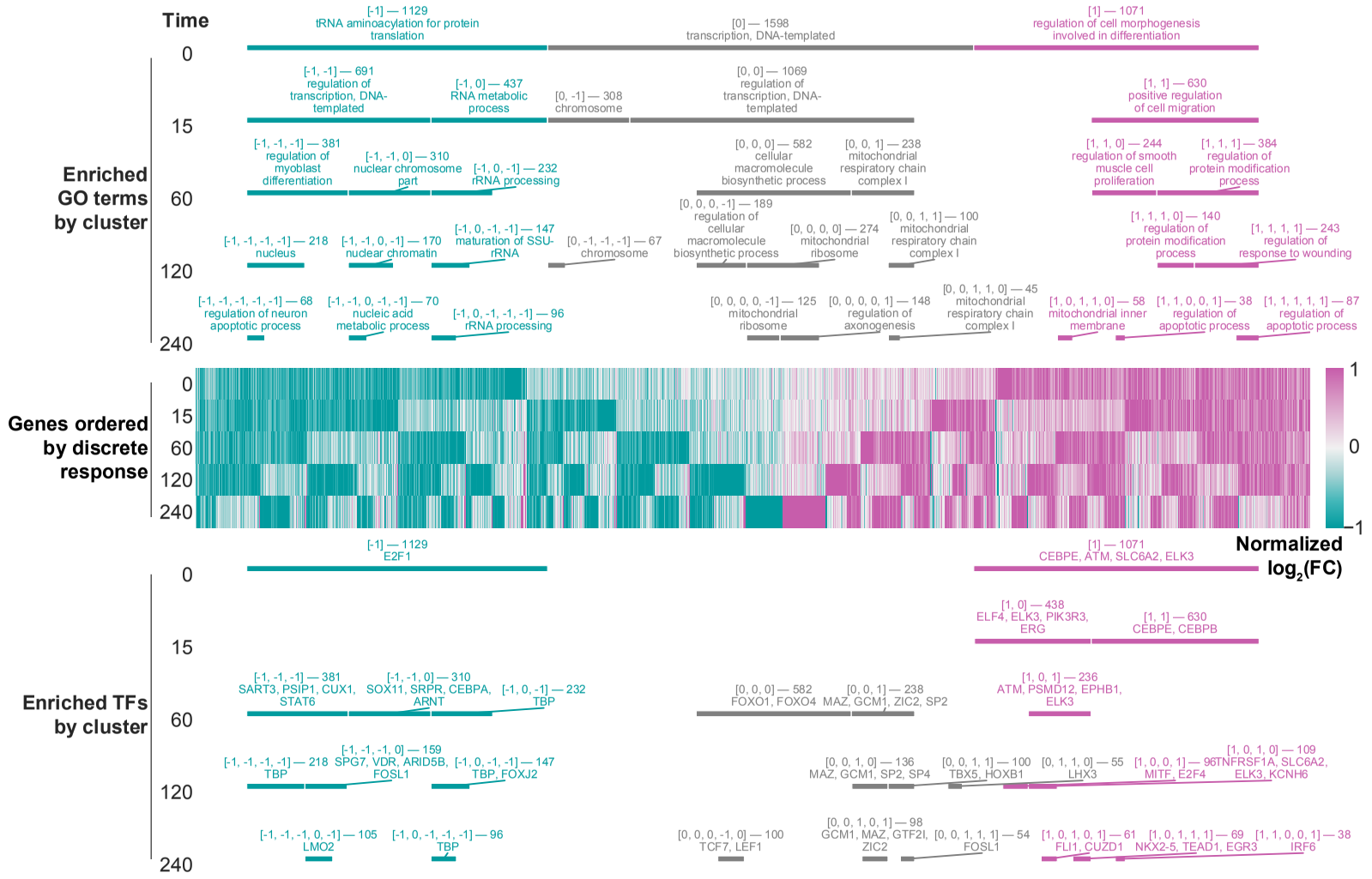

Time
Enriched GO terms by cluster
Genes ordered by discrete response
Enriched TFs by cluster
0
15
60
120
240
[-1] — 1129
tRNA aminoacylation for protein translation
[0] — 1598
transcription, DNA-templated
[1] — 1071
regulation of cell morphogenesis involved in differentiation
[-1, -1] — 691
regulation of transcription, DNA-templated
[-1, 0] — 437
RNA metabolic process
[0, -1] — 308
chromosome
[0, 0] — 1069
regulation of transcription, DNA-templated
[1, 1] — 630
positive regulation of cell migration
[-1, -1, -1] — 381
regulation of myoblast differentiation
[-1, -1, 0] — 310
nuclear chromosome part
[-1, 0, -1] — 232
rRNA processing
[0, 0, 0] — 582
cellular macromolecule biosynthetic process
[0, 0, 1] — 238
mitochondrial respiratory chain complex I
[1, 1, 0] — 244
regulation of smooth muscle cell proliferation
[1, 1, 1] — 384
regulation of protein modification process
[-1, -1, -1, -1] — 218
nucleus
[-1, -1, 0, -1] — 170
nuclear chromatin
[-1, 0, -1, -1] — 147
maturation of SSU-rRNA
[0, -1, -1, -1] — 67
chromosome
[0, 0, 0, -1] — 189
regulation of cellular macromolecule biosynthetic process
[0, 0, 0, 0] — 274
mitochondrial ribosome
[0, 0, 1, 1] — 100
mitochondrial respiratory chain complex I
[1, 1, 1, 0] — 140
regulation of protein modification process
[1, 1, 1, 1] — 243
regulation of response to wounding
[-1, -1, -1, -1, -1] — 68
regulation of neuron apoptotic process
[-1, -1, 0, -1, -1] — 70
nucleic acid metabolic process
[-1, 0, -1, -1, -1] — 96
rRNA processing
[0, 0, 0, 0, -1] — 125
mitochondrial ribosome
[0, 0, 0, 0, 1] — 148
regulation of axonogenesis
[0, 0, 1, 1, 0] — 45
mitochondrial respiratory chain complex I
[1, 0, 1, 1, 0] — 58
mitochondrial inner membrane
[1, 1, 0, 0, 1] — 38
regulation of apoptotic process
[1, 1, 1, 1, 1] — 87
regulation of apoptotic process
1
0
−1
Normalized log2(FC)
[-1] — 1129
E2F1
[1] — 1071
CEBPE, ATM, SLC6A2, ELK3
[1, 0] — 438
ELF4, ELK3, PIK3R3, ERG
[1, 1] — 630
CEBPE, CEBPB
[-1, -1, -1] — 381
SART3, PSIP1, CUX1, STAT6
[-1, -1, 0] — 310
SOX11, SRPR, CEBPA, ARNT
[-1, 0, -1] — 232
TBP
[0, 0, 0] — 582
FOXO1, FOXO4
[0, 0, 1] — 238
MAZ, GCM1, ZIC2, SP2
[1, 0, 1] — 236
ATM, PSMD12, EPHB1, ELK3
[-1, -1, -1, 0] — 159
[-1, -1, -1, -1] — 218
TBP
SPG7, VDR, ARID5B, FOSL1
[-1, 0, -1, -1] — 147
TBP, FOXJ2
[0, 0, 1, 0] — 136
MAZ, GCM1, SP2, SP4
[0, 0, 1, 1] — 100
TBX5, HOXB1
[0, 1, 1, 0] — 55
LHX3
[1, 0, 0, 1] — 96
MITF, E2F4
[1, 0, 1, 0] — 109
TNFRSF1A, SLC6A2, ELK3, KCNH6
[-1, -1, -1, 0, -1] — 105
LMO2
[-1, 0, -1, -1, -1] — 96
TBP
[0, 0, 0, -1, 0] — 100
TCF7, LEF1
[0, 0, 1, 0, 1] — 98
GCM1, MAZ, GTF2I, ZIC2
[0, 0, 1, 1, 1] — 54
FOSL1
[1, 0, 1, 0, 1] — 61
FLI1, CUZD1
[1, 0, 1, 1, 1] — 69
NKX2-5, TEAD1, EGR3
[1, 1, 0, 0, 1] — 38
IRF6

Figure 4.S3. GO and TF enrichment by discrete cluster. (top) Enriched GO terms associated with each discrete cluster possible at that time point. The discrete cluster is shown in square brackets along with the number of genes in the cluster. Clusters are colored by the sign of their LFC due to the *Spry* knock out. (middle) Heatmap of LFC values for the LFC between the *Spry* KO and WT. Values are normalized based on the maximum absolute LFC in each column. The columns are sorted by the discrete response cluster to which the genes belong. (top) Enriched TFs associated with each discrete cluster possible at that time point. The discrete cluster is shown in square brackets along with the number of genes in the cluster. Clusters are colored by the sign of their LFC due to the *Spry* knock out.

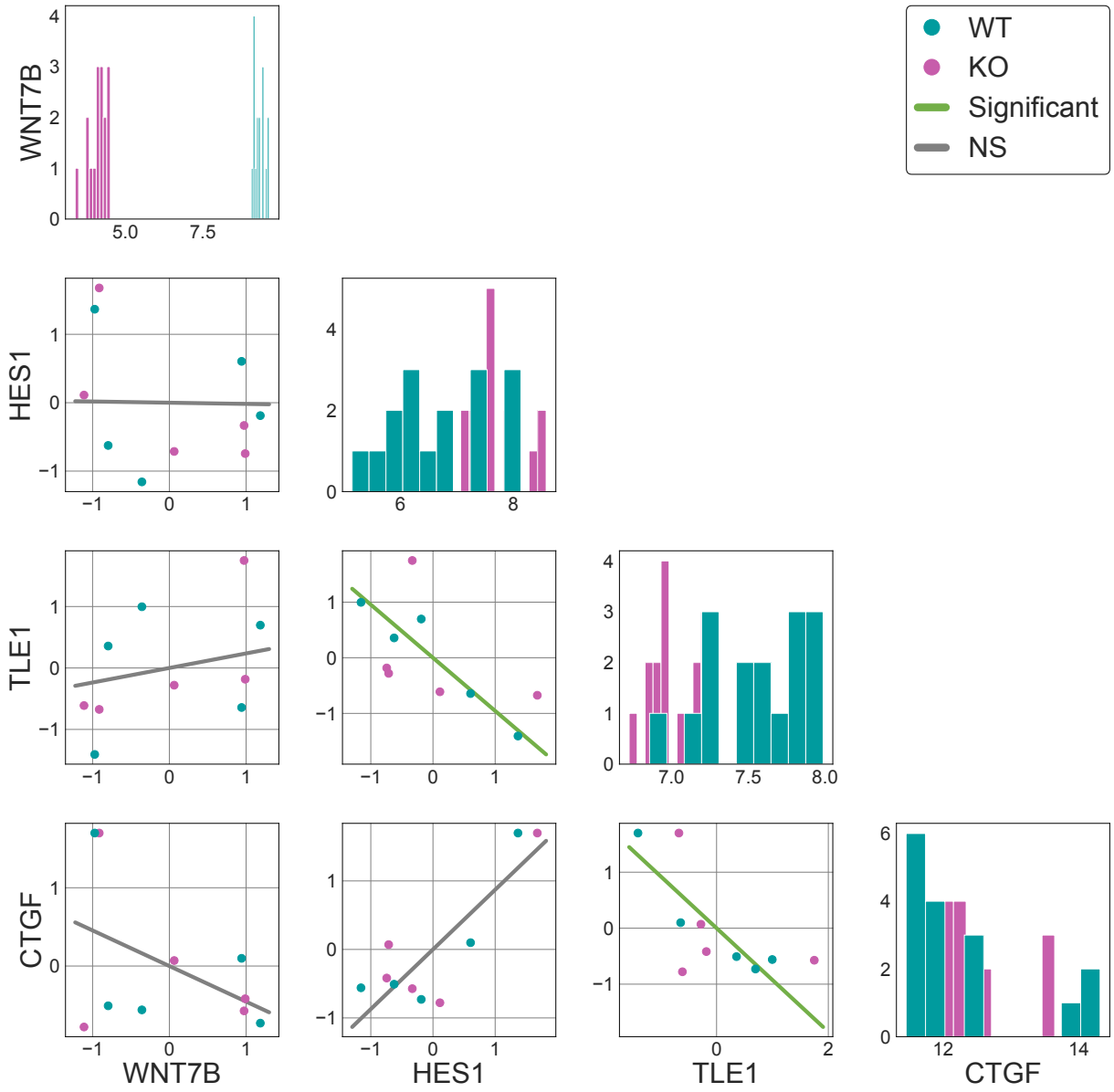


Figure 4.S4. Pairwise correlation of Wnt pathway genes and *Ctgf*. Scatter plots show the joint distributions of gene expression values in both conditions. Linear correlation lines were calculated with data points from both conditions. Significant relationship values (pearson correlation coefficient p <0.05) are highlighted in green. Histograms show the distribution of expression values between the WT and KO conditions

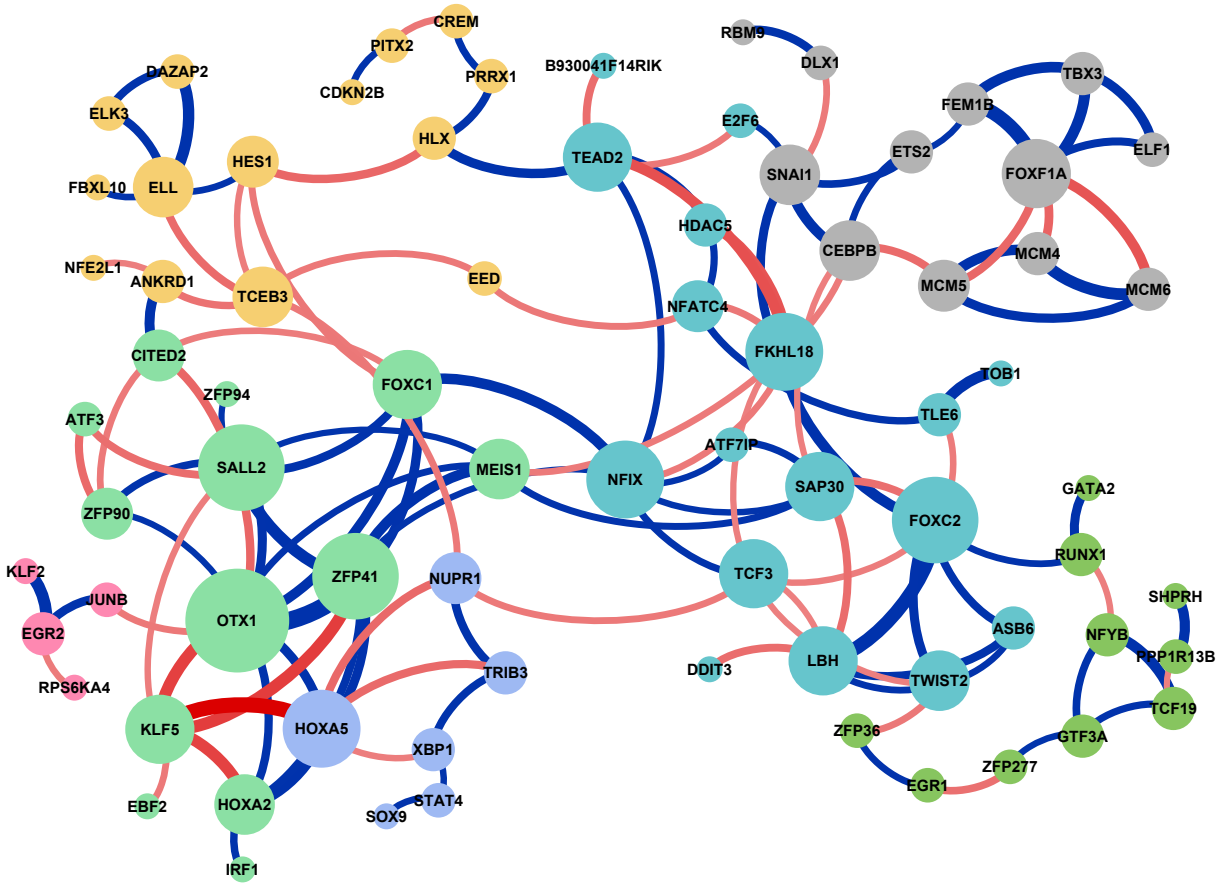


Figure 4.S5. Correlation network for DRG TFs. Node size is scaled by degree and color is assigned based on the computed module to which the node belongs. Edges are colored by the magnitude of the correlation coefficient (red=negative, blue=positive).

CHAPTER 5

Concluding remarks

In this work, I uncovered novel insights into several different biological systems by making principled use of time-series gene expression data. In Chapter 2, I showed how the SWING framework uses temporal information to better infer gene regulatory networks. These networks describe the structure, or connections, of gene regulation and are a critical step in understanding how information propagates throughout cells to actuate a response. In Chapter 3, I expanded the use of time-series expression data to create ensembles of differential equation models that produce quantitative predictions of expression in untested contexts. Training differential equation models typically requires more data than is generated by gene expression studies, but with DiffExPy I again capitalize on time-series data to bring the predictive ability of differential equation systems to many different genes. I validated the accuracy of DiffExPy and demonstrated how it uncovers biological insights using published time-series gene expression data. Lastly, in Chapter 4, I presented a top-down analysis of time-series gene expression data to understand the transcriptional role of Sprouty in response to fibroblast growth factor.

Heuristically, many current studies focus on measuring gene expression under different genetic, chemical, and environmental conditions. This experimental strategy produced significant biological findings[46,176,177]. Fewer studies exist that measure time-series gene expression. However, with careful analysis, we can use gene expression dynamics to

both qualitatively and quantitatively characterize regulatory systems. As more time-series measurements become available, I expect tools that rely on these measurements to further our understanding of biological phenomena.

5.1. Computational improvements

Were I to continue working on these projects, there are several avenues that warrant further exploration and technical improvements, . In Chapters 3 and 4, I used time-series data to gain insights at different levels of abstraction by finding enriched GO terms, transcription factors, and training ensemble models. Yet, with a genomic data set the tension always exists between identifying global trends and contextualizing the results of a specific gene or protein of interest. I discuss two strategies that could help bridge this gap of genomic scales.

5.1.1. Integration with prior knowledge databases

One challenge high-throughput data presents is contextualizing the myriad observations in a meaningful biological framework. This problem could be partially solved by integrating the output with prior knowledge databases such as KEGG, BioGRID, and StringDB[30,31,158]. Here I describe how integrating prior knowledge into the algorithms of both SWING and DiffExPy could benefit the interpretation of their results.

When developing SWING, we simplified the representation of interactions by compiling inferred edges with the same parent and child nodes but different delays into a single edge[48]. This approach fits nicely into the existing field of network inference, but it discards information about delays. I believe future work could deconvolute this temporal information to create networks that discriminate between direct and indirect edges[56]. One approach to achieve this is by merging the inferred network with prior knowledge

networks to identify directed paths between nodes. By setting assumptions about regulatory delays, an expected delay for each route between nodes in a prior knowledge network could be calculated and mapped to the observed delays. For example, if the inferred delay is short it may indicate that in the experimental context a path with fewer intermediates was activated. I suspect finding data and validating this approach would not be trivial, but could improve our understanding of context-specific pathway activation.

Computationally searching prior knowledge could also improve the interpretability of DiffExPy results. The results I presented using DiffExPy are manually selected examples that highlight use cases of the models. I spent significant time searching papers and databases to identify when my results fit with known knowledge and when they did not. Due to this constraint, I chose to focus on the angiogenesis results in Chapter 4. An automated approach would identify sets of similarly responding genes that are enriched for association with specific pathways using known gene sets[22], find the known connectivity of these pathway genes from prior knowledge, and compare the known regulatory logic with the observed gene dynamics. In fact, the cluster score used by DiffExPy was originally designed to facilitate gene rankings for gene set enrichment analysis[22].

Currently, DiffExPy is only designed to compare expression between two genetic or treatment conditions and generate models with three nodes. If three genetic conditions existed—such as the wild-type, gene A knockout, and gene B knockout—the conditions would need to be tested in pairs, with no clear way to integrate the results, and the exhaustive scan of four node networks would be computationally expensive, or prohibitive. It may instead be possible to develop a strategy to merge the trained three node ensemble models using prior knowledge networks.

5.1.2. Understanding breadth and depth with visualizations

Another strategy to bridge the understanding between genomic scales is to enable researchers to view their data in meaningful ways. How we choose to visualize data and results greatly impacts the conclusions we draw from them[178,179]. Heatmaps are popular plots used to visualize global genomic information, but they occlude information about individual samples or genes[40,176,180,181]. Conversely, scatter, line, and box plots present detailed information between variables, but only a few genes can be simultaneously compared before the plots become too confusing. Juxtaposing these plots, and creating new ones to support viewing mesoscale results of tens or hundreds of genes, could greatly help researchers interpret their results. Some genome viewers exist, but their capabilities and functionality vary greatly in quality[176,177,182].

An integrated, interactive viewer would enhance our interpretation of biological data. I envision software that enables users to visualize global trends, select genes of interest, and place the measured data in biological context. Scores of insight hides within large data sets that were only analyzed once and then left to gather dust on the GeneExpressionOmnibus[28,29]. Easy-to-use visualizations would be powerful tools to reveal information in biological data sets young and old.

5.2. Common threads

Several of the discussion points that follow were addressed in the preceding chapters. However, they are worth revisiting at a higher level in the context of time-series gene expression data.

5.2.1. Data integration

My research focused on time-series gene expression data, but there is much that can be learned using other omics technologies separately and in combination with gene expression. Gathering time-series data of both gene and protein expression could vastly improve our understanding of the link between the two[183,184]. Many of my results relied on GeneNetWeaver models that include an RNA and protein component. In each case, the protein component was ignored, as there was no experimental data with which to compare it. Simultaneously measured gene and protein time-series expression data could greatly improve the DiffExPy generated models by training models using the both components.

ChIP-seq measures the interactions between proteins and DNA, and it is a natural complement to gene expression data[185]. Identifying where a transcription factor binds relative to a gene locus and the transcription factor's subsequent impact on that gene's expression seems straightforward. However, enhancer regions have been identified up to 100 megabase pairs away from transcriptional start sites[153], so correlating peaks in ChIP-seq data with their effect on expression is difficult[185]. In practice it is challenging and expensive to collect matching omics data, and studies often end up with too few samples to statistically match ChIP- and RNA-seq data[17]. Nevertheless, as technology improves and costs decrease, I believe pairing time-series data of differential peaks from ChIP-seq[186] with differential gene expression will help clarify the role of DNA-binding events on gene expression.

5.2.2. Closing the experimental loop

There is one important omission from this work: external experimental validation. When developing the SWING and DiffExPy methods, we incorporated strong internal validation of each method's prediction accuracy. However, the biological insights derived from both methods are ultimately only hypotheses. Without additional experimental data it is impossible to prove or disprove these new hypotheses. My collaborators on these projects are fantastic, but they moved in different directions after completing their experiments. It therefore falls on the community at large to validate my findings. In an academic environment that increasingly requires experimental and computational collaboration, it is important to create a process in which experimental data is collected, computational models are generated, and then the loop is closed by validating the results. Too often the last step is not completed within the collaboration, and it is difficult for the larger research community to reproduce and use the published results.

5.2.3. Code quality in academia

I would be remiss if I did not petition for placing increased value on the *quality* of code that is produced during research. It is my estimation that the incentives of academic research currently do not encourage computational biologists to build, and importantly maintain, code that elicits complete trust in the results. This problem is not exclusive to computational research—experimental protocols are often difficult to reproduce—but the software community provides abundant tools to solve this problem. While the drive for novel, impactful results is strong, I believe we should encourage better code. What good are novel results if we cannot be confident they are true[187]? Documentation, version control, unit testing, and narrative science platforms are tools in a programmer's arsenal

to help researchers produce usable code with reproducible results[188]. New computational researchers should be taught to use these tools from the start, not apply them to code *post-hoc*, if ever. This approach may seem tedious, but I think it will help science march forward with more confidence.

5.3. Parting words

Biologists have set themselves the challenging task of studying living systems, which are orders of magnitude more complex than the most complicated of systems designed by humans[189]. We therefore need biologists of all persuasions to fully characterize and better understand biological systems. As former IBiS student Adam Hockenberry put it:

> *Interpreting* results and asking how they fit into, and/or disrupt our existing knowledge of molecules, pathways, cells, organisms, populations, and ecosystems is what makes a researcher a biologist, no matter the methods[190].

As biologists, we can and should continue to deeply interrogate individual genes and proteins. But to fully understand such complex systems we must also decipher how the parts work together to form complexes, pathways, cells, and more. I hope the work I presented here will also help biologists work together to advance our understanding of biology.

References

1. Sawyers, C. Targeted cancer therapy. *Nature* **432**, 294 (2004).

2. Pardoll, D. M. The blockade of immune checkpoints in cancer immunotherapy. *Nature Reviews Cancer* **12**, 252 (2012).

3. Hayes, J. D. & Wolf, C. R. Molecular mechanisms of drug resistance. *Biochemical Journal* **272**, 281 (1990).

4. Longley, D. & Johnston, P. Molecular mechanisms of drug resistance. *The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland* **205**, 275–292 (2005).

5. Pisco, A. O. *et al.* Non-darwinian dynamics in therapy-induced cancer drug resistance. *Nature communications* **4**, 2467 (2013).

6. Rieth, J. & Subramanian, S. Mechanisms of intrinsic tumor resistance to immunotherapy. *International journal of molecular sciences* **19**, 1340 (2018).

7. Zhang, C. *et al.* Raf inhibitors that evade paradoxical mapk pathway activation. *Nature* **526**, 583 (2015).

8. Suntharalingam, G. *et al.* Cytokine storm in a phase 1 trial of the anti-cd28 monoclonal antibody tgn1412. *New England Journal of Medicine* **355**, 1018–1028 (2006).

9. Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L. & Wold, B. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods* **5**, 621 (2008).

10. Johnson, D. S., Mortazavi, A., Myers, R. M. & Wold, B. Genome-wide mapping of in vivo protein-dna interactions. *Science* **316**, 1497–1502 (2007).

11. Koek, M. M., Jellema, R. H., van der Greef, J., Tas, A. C. & Hankemeier, T. Quantitative metabolomics based on gas chromatography mass spectrometry: status and perspectives. *Metabolomics* **7**, 307–328 (2011).

12. Aebersold, R. & Mann, M. Mass spectrometry-based proteomics. *Nature* **422**, 198 (2003).

13. Crawford, G. E. *et al.* Dnase-chip: a high-resolution method to identify dnase i hypersensitive sites using tiled microarrays. *Nature methods* **3**, 503 (2006).

14. Buenrostro, J. D., Wu, B., Chang, H. Y. & Greenleaf, W. J. Atac-seq: a method for assaying chromatin accessibility genome-wide. *Current protocols in molecular biology* **109**, 21–29 (2015).

15. Schena, M., Shalon, D., Davis, R. W. & Brown, P. O. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science* **270**, 467–470 (1995).

16. Hashimshony, T., Wagner, F., Sher, N. & Yanai, I. Cel-seq: single-cell rna-seq by multiplexed linear amplification. *Cell reports* **2**, 666–673 (2012).

17. Hafner, A. *et al.* p53 pulses lead to distinct patterns of gene expression albeit similar dna-binding dynamics. *Nature Structural and Molecular Biology* **24**, 840 (2017).

18. Lim, L. P. *et al.* Microarray analysis shows that some micrornas downregulate large numbers of target mrnas. *Nature* **433**, 769 (2005).

19. Trapnell, C. *et al.* Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology* **28**, 511 (2010).

20. Gaublomme, J. T. *et al.* Single-cell genomics unveils critical regulators of th17 cell pathogenicity. *Cell* **163**, 1400–1412 (2015).

21. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. *Nature genetics* **25**, 25 (2000).

22. Subramanian, A. *et al.* Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences* **102**, 15545–15550 (2005).

23. Mi, H. *et al.* Panther version 11: expanded annotation data from gene ontology and reactome pathways, and data analysis tool enhancements. *Nucleic acids research* **45**, D183–D189 (2016).

24. Consortium, G. O. Expansion of the gene ontology knowledgebase and resources. *Nucleic acids research* **45**, D331–D338 (2016).

25. Kiselev, V. Y. *et al.* Perturbations of pip3 signalling trigger a global remodelling of mrna landscape and reveal a transcriptional feedback loop. *Nucleic acids research* **43**, 9663–9679 (2015).

26. Janes, K. A. *et al.* A systems model of signaling identifies a molecular basis set for cytokine-induced apoptosis. *Science* **310**, 1646–1653 (2005).

27. Ciaccio, M. F., Finkle, J. D., Xue, A. Y. & Bagheri, N. A systems approach to integrative biology: an overview of statistical methods to elucidate association and architecture. *Integrative and comparative biology* icu037 (2014).

28. Edgar, R., Domrachev, M. & Lash, A. E. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research* **30**, 207–210 (2002).

29. Barrett, T. *et al.* Ncbi geo: archive for functional genomics data sets—update. *Nucleic acids research* **41**, D991–D995 (2012).

30. Chatr-Aryamontri, A. *et al.* The biogrid interaction database: 2017 update. *Nucleic acids research* **45**, D369–D379 (2017).

31. Szklarczyk, D. *et al.* String v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic acids research* **43**, D447–D452 (2014).

32. Kitano, H. Computational systems biology. *Nature* **420**, 206 (2002).

33. Amaral, L. A. N. A truer measure of our ignorance. *Proceedings of the National Academy of Sciences* **105**, 6795–6796 (2008).

34. Howson, C. & Urbach, P. *Scientific reasoning: the Bayesian approach* (Open Court Publishing, 2006).

35. Fischer, H. P. Mathematical modeling of complex biological systems: from parts lists to understanding systems behavior. *Alcohol Research & Health* **31**, 49 (2008).

36. Janes, K. A. *et al.* Cue-signal-response analysis of tnf-induced apoptosis by partial least squares regression of dynamic multivariate data. *Journal of Computational Biology* **11**, 544–561 (2004).

37. Aldridge, B. B., Burke, J. M., Lauffenburger, D. A. & Sorger, P. K. Physicochemical modelling of cell signalling pathways. *Nature cell biology* **8**, 1195 (2006).

38. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–140 (2010).

39. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic rna-seq quantification. *Nature biotechnology* **34**, 525 (2016).

40. Langfelder, P. & Horvath, S. Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics* **9**, 559 (2008).

41. Pearson, K. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2**, 559–572 (1901).

42. Amir, E.-a. D. *et al.* visne enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature biotechnology* **31**, 545 (2013).

43. Platzer, A. Visualization of snps with t-sne. *PloS one* **8**, e56883 (2013).

44. Ritchie, M. E. *et al.* limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research* **43**, e47 (2015).

45. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology* **15**, 550 (2014).

46. Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A. & Nolan, G. P. Causal protein-signaling networks derived from multiparameter single-cell data. *Science* **308**, 523–529 (2005).

47. Marbach, D. *et al.* Wisdom of crowds for robust gene network inference. *Nature methods* **9**, 796–804 (2012).

48. Finkle, J. D., Wu, J. J. & Bagheri, N. Windowed granger causal inference strategy improves discovery of gene regulatory networks. *Proceedings of the National Academy of Sciences* **115**, 2252–2257 (2018).

49. Fisher, R. A. On the interpretation of χ 2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society* **85**, 87–94 (1922).

50. Massey Jr, F. J. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association* **46**, 68–78 (1951).

51. Ciaccio, M. F., Chen, V. C., Jones, R. B. & Bagheri, N. The dionesus algorithm provides scalable and accurate reconstruction of dynamic phosphoproteomic networks to reveal new drug targets. *Integrative Biology* **7**, 776–791 (2015).

52. Yu, J., Xue, A., Redei, E. & Bagheri, N. A support vector machine model provides an accurate transcript-level-based diagnostic for major depressive disorder. *Translational psychiatry* **6**, e931 (2016).

53. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288 (1996).

54. Haury, A.-C., Mordelet, F., Vera-Licona, P. & Vert, J.-P. Tigress: trustful inference of gene regulation using stability selection. *BMC systems biology* **6**, 145 (2012).

55. Huynh-Thu, V. A., Irrthum, A., Wehenkel, L. & Geurts, P. Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. *PLoS ONE* **5**, e12776 (2010).

56. Feizi, S., Marbach, D., Médard, M. & Kellis, M. Network deconvolution as a general method to distinguish direct dependencies in networks. *Nature biotechnology* **31**, 726 (2013).

57. Ryu, H. *et al.* Frequency modulation of erk activation dynamics rewires cell fate. *Molecular systems biology* **11**, 838 (2015).

58. Chu, L.-H. *et al.* A multiscale computational model predicts distribution of anti-angiogenic isoform vegf 165b in peripheral arterial disease in human and mouse. *Scientific reports* **6**, 37030 (2016).

59. Ma, W., Trusina, A., El-Samad, H., Lim, W. A. & Tang, C. Defining network topologies that can achieve biochemical adaptation. *Cell* **138**, 760–773 (2009).

60. Hartfield, R. M., Schwarz, K. A., Muldoon, J. J., Bagheri, N. & Leonard, J. N. Multiplexing engineered receptors for multiparametric evaluation of environmental ligands. *ACS synthetic biology* **6**, 2042–2055 (2017).

61. Schaffter, T., Marbach, D. & Floreano, D. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics* **27**, 2263–2270 (2011).

62. Neuert, G. *et al.* Systematic identification of signal-activated stochastic gene regulation. *Science* **339**, 584–587 (2013).

63. Karr, J. R. *et al.* A whole-cell computational model predicts phenotype from genotype. *Cell* **150**, 389–401 (2012).

64. Sulaimanov, N., Klose, M., Busch, H. & Boerries, M. Understanding the mtor signaling pathway via mathematical modeling. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* **9**, e1379 (2017).

65. Gambin, A., Charzyńska, A., Ellert-Miklaszewska, A. & Rybiński, M. Computational models of the jak1/2-stat1 signaling. *Jak-stat* **2**, e24672 (2013).

66. Orton, R. J. *et al.* Computational modelling of the receptor-tyrosine-kinase-activated mapk pathway. *Biochemical Journal* **392**, 249–261 (2005).

67. Pappalardo, F. *et al.* Computational modeling of pi3k/akt and mapk signaling pathways in melanoma cancer. *PLoS One* **11**, e0152104 (2016).

68. Sakamoto, E. & Iba, H. Inferring a system of differential equations for a gene regulatory network by using genetic programming. In *Proc. congress on evolutionary computation*, vol. 1, 720–726 (2001).

69. Mangan, N. M., Brunton, S. L., Proctor, J. L. & Kutz, J. N. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications* **2**, 52–63 (2016).

70. Mangan, N. M., Kutz, J. N., Brunton, S. L. & Proctor, J. L. Model selection for dynamical systems via sparse regression and information criteria. *Proc. R. Soc. A* **473**, 20170009 (2017).

71. François, P. & Hakim, V. Design of genetic networks with specified functions by evolution in silico. *Proceedings of the National Academy of Sciences of the United States of America* **101**, 580–585 (2004).

72. Nagoshi, E. *et al.* Circadian Gene Expression in Individual Fibroblasts: Cell-Autonomous and Self-Sustained Oscillators Pass Time to Daughter Cells. *Cell* **119**, 693–705 (2004).

73. Spellman, P. T. *et al.* Comprehensive Identification of Cell Cycle–regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. *Molecular Biology of the Cell* **9**, 3273–3297 (1998).

74. Geva-Zatorsky, N. *et al.* Oscillations and variability in the p53 system. *Molecular Systems Biology* **2**, 2006.0033 (2006).

75. Jiang, Y.-J. *et al.* Notch signalling and the synchronization of the somite segmentation clock. *Nature* **408**, 475–479 (2000).

76. Madar, A., Greenfield, A., Ostrer, H., Vanden-Eijnden, E. & Bonneau, R. The Inferelator 2.0: a scalable framework for reconstruction of dynamic regulatory network models. *Conference proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference* **2009**, 5448–5451 (2009).

77. van Someren, E. P. *et al.* Least absolute regression network analysis of the murine osteoblast differentiation network. *Bioinformatics (Oxford, England)* **22**, 477–484

(2006).

78. Äijö, T., Granberg, K. & Lähdesmäki, H. Sorad: a systems biology approach to predict and modulate dynamic signaling pathway response from phosphoproteome time-course measurements. *Bioinformatics (Oxford, England)* **29**, 1283–1291 (2013).

79. Zak, D. E., Gonye, G. E., Schwaber, J. S. & Doyle, F. J. Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network. *Genome Research* **13**, 2396–2405 (2003).

80. Raue, A., Becker, V., Klingmüller, U. & Timmer, J. Identifiability and observability analysis for experimental design in nonlinear dynamical models. *Chaos (Woodbury, N.Y.)* **20**, 045105 (2010).

81. Lawrence, N. D., Sanguinetti, G. & Rattray, M. Modelling transcriptional regulation using Gaussian Processes. In Schölkopf, P. B., Platt, J. C. & Hoffman, T. (eds.) *Advances in Neural Information Processing Systems 19*, 785–792 (MIT Press, 2007).

82. Brunton, S. L., Proctor, J. L. & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences* **113**, 3932–3937 (2016).

83. Huynh-Thu, V. A. & Sanguinetti, G. Combining tree-based and dynamical systems for the inference of gene regulatory networks. *Bioinformatics (Oxford, England)* **31**, 1614–1622 (2015).

84. Granger, C. W. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society* 424–438 (1969).

85. Lozano, A. C., Abe, N., Liu, Y. & Rosset, S. Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics* **25**, i110–i118 (2009).

86. Zoppoli, P., Morganella, S. & Ceccarelli, M. Timedelay-aracne: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC bioinformatics* **11**, 154 (2010).

87. Shojaie, A. & Michailidis, G. Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics* **26**, i517–i523 (2010).

88. Petralia, F., Wang, P., Yang, J. & Tu, Z. Integrative random forest for gene regulatory network inference. *Bioinformatics (Oxford, England)* **31**, i197–205 (2015).

89. Quinn, C. J., Coleman, T. P., Kiyavash, N. & Hatsopoulos, N. G. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *Journal of Computational Neuroscience* **30**, 17–44 (2011).

90. Gedeon, T. & Bokes, P. Delayed Protein Synthesis Reduces the Correlation between mRNA and Protein Fluctuations. *Biophysical Journal* **103**, 377–385 (2012).

91. McAdams, H. H. & Arkin, A. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences* **94**, 814–819 (1997).

92. Ronen, M., Rosenberg, R., Shraiman, B. I. & Alon, U. Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. *Proceedings of the national academy of sciences* **99**, 10555–10560 (2002).

93. Bansal, M., Della Gatta, G. & di Bernardo, D. Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics (Oxford, England)* **22**, 815–822 (2006).

94. Morshed, N., Chetty, M. & Xuan Vinh, N. Simultaneous learning of instantaneous and time-delayed genetic interactions using novel information theoretic scoring technique. *BMC Systems Biology* **6**, 62 (2012).

95. Boker, S. M., Xu, M., Rotondo, J. L. & King, K. Windowed cross-correlation and peak picking for the analysis of variability in the association between behavioral time series. *Psychological Methods* **7**, 338–355 (2002).

96. Bader, G. D. & Hogue, C. W. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* **4**, 2 (2003).

97. Ganduri, Y. L., Sadda, S. R., Datta, M. W., Jambukeswaran, R. K. & Datta, P. Tdca, a transcriptional activator of the tdcabc operon of escherichia coli, is a member of the lysr family of proteins. *Molecular & general genetics: MGG* **240**, 395–402 (1993).

98. Shimada, T., Fujita, N., Yamamoto, K. & Ishihama, A. Novel roles of camp receptor protein (crp) in regulation of transport and metabolism of carbon sources. *PLOS ONE* **6**, e20081 (2011).

99. Crack, J., Green, J. & Thomson, A. J. Mechanism of oxygen sensing by the bacterial transcription factor fumarate-nitrate reduction (fnr). *The Journal of Biological Chemistry* **279**, 9278–9286 (2004).

100. Kao, K. C., Tran, L. M. & Liao, J. C. A Global Regulatory Role of Gluconeogenic Genes in Escherichia coli Revealed by Transcriptome Network Analysis. *Journal of Biological Chemistry* **280**, 36079–36087 (2005).

101. Zhang, J. *et al.* Lysine Acetylation Is a Highly Abundant and Evolutionarily Conserved Modification in Escherichia Coli. *Molecular & Cellular Proteomics : MCP* **8**, 215–225 (2009).

102. Runge, J. *et al.* Identifying causal gateways and mediators in complex spatio-temporal systems. *Nature Communications* **6**, ncomms9502 (2015).

103. Studham, M. E., Tjärnberg, A., Nordling, T. E., Nelander, S. & Sonnhammer, E. L. L. Functional association networks as priors for gene regulatory network inference. *Bioinformatics* **30**, i130–i138 (2014).

104. Hill, S. M. *et al.* Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature Methods* **13**, 310–318 (2016).

105. Irrthum, A., Wehenkel, L., Geurts, P. *et al.* Inferring regulatory networks from expression data using tree-based methods. *PloS one* **5**, e12776 (2010).

106. Marbach, D. *et al.* Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the national academy of sciences* **107**, 6286–6291 (2010).

107. Weisstein, E. W. Bonferroni Correction (2017).

108. Gama-Castro, S. *et al.* RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic Acids Research* **44**, D133–143 (2016).

109. Klopfenstein, D. *et al.* Goatools v0.6.10. *Zenodo* (2016).

110. Interpolation (scipy.interpolate) — scipy v0.19.0 reference guide (2017).

111. Walt, S. v. d., Colbert, S. C. & Varoquaux, G. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering* **13**, 22–30 (2011).

112. McKinney, W. Data structures for statistical computing in python. In van der Walt, S. & Millman, J. (eds.) *Proceedings of the 9th Python in Science Conference*, 51 – 56 (2010).

113. Hagberg, A. A., Schult, D. A. & Swart, P. J. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, 11–15 (Pasadena, CA USA, 2008).

114. Pedregosa, F. *et al.* Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

115. Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science Engineering* **9**, 90–95 (2007).

116. Jozefczuk, S. *et al.* Metabolomic and transcriptomic stress response of Escherichia coli. *Molecular Systems Biology* **6**, 364 (2010).

117. Faith, J. J. *et al.* Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology* **5**, e8 (2007).

118. Shalem, O. *et al.* Transient transcriptional responses to stress are generated by opposing effects of mRNA production and degradation. *Molecular Systems Biology* **4**, 223 (2008).

119. Li, C. M. & Klevecz, R. R. A rapid genome-scale response of the transcriptional oscillator to perturbation reveals a period-doubling path to phenotypic change. *Proceedings of the National Academy of Sciences of the United States of America* **103**, 16254–16259 (2006).

120. Orlando, D. A. *et al.* Global control of cell-cycle transcription by coupled CDK and network oscillators. *Nature* **453**, 944–947 (2008).

121. Finkle, J. D. & Bagheri, N. Hybrid analysis of gene dynamics predicts context specific expression and offers regulatory insights. *Bioinformatics* (2019).

122. Emilsson, V. *et al.* Genetics of gene expression and its effect on disease. *Nature* **452**, 423 (2008).

123. Law, C. W., Chen, Y., Shi, W. & Smyth, G. K. voom: Precision weights unlock linear model analysis tools for rna-seq read counts. *Genome biology* **15**, R29 (2014).

124. Pimentel, H., Bray, N. L., Puente, S., Melsted, P. & Pachter, L. Differential analysis of rna-seq incorporating quantification uncertainty. *Nature methods* **14**, 687 (2017).

125. Jiang, Y.-J. *et al.* Notch signalling and the synchronization of the somite segmentation clock. *Nature* **408**, 475 (2000).

126. Jiang, P. *et al.* A systems approach identifies networks and genes linking sleep and stress: implications for neuropsychiatric disorders. *Cell reports* **11**, 835–848 (2015).

127. Schulz, M. H. *et al.* Drem 2.0: Improved reconstruction of dynamic regulatory networks from time-series expression data. *BMC systems biology* **6**, 104 (2012).

128. Spies, D., Renz, P. F., Beyer, T. A. & Ciaudo, C. Comparative analysis of differential gene expression tools for rna sequencing time course data. *Briefings in bioinformatics* (2017).

129. Madar, A., Greenfield, A., Ostrer, H., Vanden-Eijnden, E. & Bonneau, R. The inferelator 2.0: a scalable framework for reconstruction of dynamic regulatory network models. In *Conf Proc IEEE Eng Med Biol Soc*, vol. 31, 5448–5451 (2009).

130. Mina, M. *et al.* Promoter-level expression clustering identifies time development of transcriptional regulatory cascades initiated by erbb receptors in breast cancer cells. *Scientific reports* **5**, 11999 (2015).

131. Lansbergen, G. *et al.* Clasps attach microtubule plus ends to the cell cortex through a complex with ll5β. *Developmental cell* **11**, 21–32 (2006).

132. Srinivas, H. *et al.* Akt phosphorylates and suppresses the transactivation of retinoic acid receptor α. *Biochemical Journal* **395**, 653–662 (2006).

133. Wise, A. & Bar-Joseph, Z. cdrem: inferring dynamic combinatorial gene regulation. *Journal of Computational Biology* **22**, 324–333 (2015).

134. Comet, I., Riising, E. M., Leblanc, B. & Helin, K. Maintaining cell identity: Prc2-mediated regulation of transcription and cancer. *Nature Reviews Cancer* **16**, 803 (2016).

135. Squazzo, S. L. *et al.* Suz12 binds to silenced regions of the genome in a cell-type-specific manner. *Genome research* **16**, 890–900 (2006).

136. Bernardo, G. M. *et al.* Foxa1 represses the molecular phenotype of basal breast cancer cells. *Oncogene* **32**, 554 (2013).

137. Potter, A. S., Casa, A. J. & Lee, A. V. Forkhead box a1 (foxa1) is a key mediator of insulin-like growth factor i (igf-i) activity. *Journal of cellular biochemistry* **113**, 110–121 (2012).

138. Riising, E. M., Boggio, R., Chiocca, S., Helin, K. & Pasini, D. The polycomb repressive complex 2 is a potential target of sumo modifications. *PLoS One* **3**, e2704 (2008).

139. Stoeger, T., Gerlach, M., Morimoto, R. I. & Amaral, L. A. N. Large-scale investigation of the reasons why potentially important genes are ignored. *PLoS biology* **16**, e2006643 (2018).

140. Tegla, C. A. *et al.* Rgc-32 is a novel regulator of the t-lymphocyte cell cycle. *Experimental and molecular pathology* **98**, 328–337 (2015).

141. Arnold, T. B. & Emerson, J. W. Nonparametric goodness-of-fit tests for discrete null distributions. *R Journal* **3**, 34–39 (2011).

142. McKinney, W. pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing* 1–9 (2011).

143. Klopfenstein, D. *et al.* Goatools: A python library for gene ontology analyses. *Scientific reports* **8**, 10872 (2018).

144. Hacohen, N., Kramer, S., Sutherland, D., Hiromi, Y. & Krasnow, M. A. sprouty encodes a novel antagonist of fgf signaling that patterns apical branching of the drosophila airways. *Cell* **92**, 253–263 (1998).

145. Kramer, S., Okabe, M., Hacohen, N., Krasnow, M. A. & Hiromi, Y. Sprouty: a common antagonist of fgf and egf signaling pathways in drosophila. *Development* **126**, 2515–2525 (1999).

146. Minowada, G. *et al.* Vertebrate sprouty genes are induced by fgf signaling and can cause chondrodysplasia when overexpressed. *Development* **126**, 4465–4475 (1999).

147. Kim, H. J. & Bar-Sagi, D. Modulation of signalling by sprouty: a developing story. *Nature reviews Molecular cell biology* **5**, 441 (2004).

148. Lim, J. *et al.* The cysteine-rich sprouty translocation domain targets mitogen-activated protein kinase inhibitory proteins to phosphatidylinositol 4, 5-bisphosphate in plasma membranes. *Molecular and cellular biology* **22**, 7953–7966 (2002).

149. Cabrita, M. A. & Christofori, G. Sprouty proteins, masterminds of receptor tyrosine kinase signaling. *Angiogenesis* **11**, 53–62 (2008).

150. Felfly, H. & Klein, O. D. Sprouty genes regulate proliferation and survival of human embryonic stem cells. *Scientific reports* **3**, 2277 (2013).

151. Fong, C. W. *et al.* Sprouty 2, an inhibitor of mitogen-activated protein kinase signaling, is down-regulated in hepatocellular carcinoma. *Cancer research* **66**, 2048–2058 (2006).

152. Lo, T. L. *et al.* The ras/mitogen-activated protein kinase pathway inhibitor and likely tumor suppressor proteins, sprouty 1 and sprouty 2 are deregulated in breast cancer. *Cancer research* **64**, 6127–6136 (2004).

153. Masoumi-Moghaddam, S., Amini, A. & Morris, D. L. The developing story of sprouty and cancer. *Cancer and Metastasis Reviews* **33**, 695–720 (2014).

154. Nabet, B. *et al.* Deregulation of the ras-erk signaling axis modulates the enhancer landscape. *Cell reports* **12**, 1300–1313 (2015).

155. Tennis, M. A. *et al.* Sprouty-4 inhibits transformed cell growth, migration and invasion, and epithelial-mesenchymal transition, and is regulated by wnt7a through pparγ in non–small cell lung cancer. *Molecular Cancer Research* 1541–7786 (2010).

156. Sabatel, C. *et al.* Sprouty1, a new target of the angiostatic agent 16k prolactin, negatively regulates angiogenesis. *Molecular cancer* **9**, 231 (2010).

157. Amit, I. *et al.* A module of negative feedback regulators defines growth factor signaling. *Nature genetics* **39**, 503 (2007).

158. Kanehisa, M. & Goto, S. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research* **28**, 27–30 (2000).

159. Casci, T., Vinós, J. & Freeman, M. Sprouty, an intracellular inhibitor of ras signaling. *Cell* **96**, 655–665 (1999).

160. Lee, S. H., Schloss, D. J., Jarvis, L., Krasnow, M. A. & Swain, J. L. Inhibition of angiogenesis by a mouse sprouty protein. *Journal of Biological Chemistry* **276**, 4128–4133 (2001).

161. Fei, Y., Xiao, L., Doetschman, T., Coffin, D. J. & Hurley, M. M. Fgf2 stimulation of osteoblast differentiation and bone formation is mediated by modulation of the wnt pathway. *Journal of Biological Chemistry* jbc–M111 (2011).

162. Mansukhani, A., Ambrosetti, D., Holmes, G., Cornivelli, L. & Basilico, C. Sox2 induction by fgf and fgfr2 activating mutations inhibits wnt signaling and osteoblast differentiation. *The Journal of cell biology* **168**, 1065–1076 (2005).

163. Bradham, D. M., Igarashi, A., Potter, R. L. & Grotendorst, G. R. Connective tissue growth factor: a cysteine-rich mitogen secreted by human vascular endothelial cells is related to the src-induced immediate early gene product cef-10. *The Journal of cell biology* **114**, 1285–1294 (1991).

164. Finger, E. *et al.* Ctgf is a therapeutic target for metastatic melanoma. *Oncogene* **33**, 1093 (2014).

165. Brigstock, D. R. Regulation of angiogenesis and endothelial cell function by connective tissue growth factor (ctgf) and cysteine-rich 61 (cyr61). *Angiogenesis* **5**, 153–165

(2002).

166. Kay, S. K. *et al.* The role of the hes1 crosstalk hub in notch-wnt interactions of the intestinal crypt. *PLoS computational biology* **13**, e1005400 (2017).

167. Diepenbruck, M. *et al.* Tead2 expression levels control yap/taz subcellular distribution, zyxin expression, and epithelial-mesenchymal transition. *J Cell Sci* jcs–139865 (2014).

168. Kim, M. & Jho, E.-h. Cross-talk between wnt/β-catenin and hippo signaling pathways: a brief review. *BMB reports* **47**, 540 (2014).

169. Dawes, L., Shelley, E., McAvoy, J. & Lovicu, F. A role for hippo/yap-signaling in fgf-induced lens epithelial cell proliferation and fibre differentiation. *Experimental eye research* **169**, 122–133 (2018).

170. Hobbs, G. A., Der, C. J. & Rossman, K. L. Ras isoforms and mutations in cancer at a glance. *J Cell Sci* **129**, 1287–1292 (2016).

171. Jeong, W.-J., Ro, E. J. & Choi, K.-Y. Interaction between wnt/β-catenin and ras-erk pathways and an anti-cancer strategy via degradations of β-catenin and ras by targeting the wnt/β-catenin pathway. *NPJ precision oncology* **2**, 5 (2018).

172. Kanamori, M. *et al.* A genome-wide and nonredundant mouse transcription factor database. *Biochemical and biophysical research communications* **322**, 787–793 (2004).

173. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)* 289–300 (1995).

174. Bastian, M., Heymann, S., Jacomy, M. *et al.* Gephi: an open source software for exploring and manipulating networks. *Icwsm* **8**, 361–362 (2009).

175. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* **2008**, P10008 (2008).

176. Kundaje, A. *et al.* Integrative analysis of 111 reference human epigenomes. *Nature* **518**, 317 (2015).

177. Weinstein, J. N. *et al.* The cancer genome atlas pan-cancer analysis project. *Nature genetics* **45**, 1113 (2013).

178. Krzywinski, M. & Altman, N. Points of significance: error bars (2013).

179. Krzywinski, M. & Altman, N. Points of significance: visualizing samples with box plots (2014).

180. Gu, Z., Eils, R., Schlesner, M. & Ishaque, N. Enrichedheatmap: an r/bioconductor package for comprehensive visualization of genomic signal associations. *BMC genomics* **19**, 234 (2018).

181. Wu, H.-M. *et al.* Covariate-adjusted heatmaps for visualizing biological data via correlation decomposition. *Bioinformatics* **1**, 10 (2018).

182. Duan, Q. *et al.* Lincs canvas browser: interactive web app to query, browse and interrogate lincs l1000 gene expression signatures. *Nucleic acids research* **42**, W449–W460 (2014).

183. Vogel, C. *et al.* Sequence signatures and mrna concentration can explain two-thirds of protein abundance variation in a human cell line. *Molecular systems biology* **6**, 400 (2010).

184. Rey, G. *et al.* Metabolic oscillations on the circadian time scale in drosophila cells lacking clock genes. *Molecular systems biology* **14**, e8376 (2018).

185. Jiang, S. & Mortazavi, A. Integrating chip-seq with other functional genomics data. *Briefings in functional genomics* **17**, 104–115 (2018).

186. Steinhauser, S., Kurzawa, N., Eils, R. & Herrmann, C. A comprehensive comparison of tools for differential chip-seq analysis. *Briefings in bioinformatics* **17**, 953–966 (2016).

187. Baker, M. Over half of psychology studies fail reproducibility test. *Nature News* (2015).

188. Arkin, A. P. *et al.* Kbase: The united states department of energy systems biology knowledgebase. *Nature biotechnology* **36** (2018).

189. Amaral, L. A. & Ottino, J. M. Complex networks. *The European Physical Journal B* **38**, 147–162 (2004).

190. Hockenberry, A. J. *Sequence Determinants of Translation Efficiency.* Ph.D. thesis, Northwestern University (2017).