A systematic comparison of computational methods for expression forecasting

Eric Kernfeld, ekernfe1@jh.edu, https://orcid.org/0000-0002-2310-8191
Yunxiao Yang, yyang117@jh.edu, https://orcid.org/0000-0003-3429-3174
Joshua S. Weinstock, jweins17@jhu.edu, https://orcid.org/0000-0001-7013-1899
Alexis Battle, ajbattle@jhu.edu (corresponding author), https://orcid.org/0000-0002-5287-627X
Patrick Cahan, patrick.cahan@jhmi.edu (corresponding author), https://orcid.org/0000-0003-3652-2540

Department of Biomedical Engineering, Johns Hopkins University
3400 N. Charles Street
Wyman Park Building
Suite 400 West
Baltimore, MD, 21218, USA

Keywords:

Transcriptional regulation, transcription factor, network inference, gene regulatory network, perturb-seq, expression prediction, expression forecasting

Abstract **3**
Introduction **4**
Results **5**
Large-scale perturbation datasets display a high rate of success in targeting individual genes, modest correlation across replicates, and mostly small global effects 5
Table 1 5
Figure S1 7
Figure 2 9
Regression-based analyses do not out-perform uninformed baselines 10
Figure 3 12
Figure S2 14
Table 4 16
Diverse published methods do not out-perform simple baselines 17
Figure 5 18
A grammar of gene regulatory network (GRN) inference synthesizes diverse methods within an extensible benchmarking framework 20
Discussion **21**
Methods **23**
Data and code availability 23
Simulated data 23
Perturbation data acquisition and preprocessing 24
Network acquisition and preprocessing 25
GRN inference methods 25
Evaluations 26
Author contributions **26**
Acknowledgements **26**
Declaration of interests **27**
Supplemental information **27**
Bibliography **27**

Abstract

Due to the abundance of single cell RNA-seq data, a number of methods for predicting expression after perturbation have recently been published. Expression prediction methods are enticing because they promise to answer pressing questions in fields ranging from developmental genetics to cell fate engineering and because they are faster, cheaper, and higher-throughput than their experimental counterparts. However, the absolute and relative accuracy of these methods is poorly characterized, limiting their informed use, their improvement, and the interpretation of their predictions. To address these issues, we created a benchmarking platform that combines a panel of large-scale perturbation datasets with an expression forecasting software engine that encompasses or interfaces to current methods. We used our platform to systematically assess methods, parameters, and sources of auxiliary data. We found that uninformed baseline predictions, which were not always included in prior evaluations, yielded the same or better mean absolute error than benchmarked methods in all test cases. These results cast doubt on the ability of current expression forecasting methods to provide mechanistic insights or to rank hypotheses for experimental follow-up. However, given the rapid pace of innovation in the field, new approaches may yield more accurate expression predictions. Our platform will serve as a neutral benchmark to improve methods and to identify contexts in which expression prediction can succeed.

Introduction

Systematically determining causal effects of transcription factor (TF) activity is a fundamental problem in systems biology, with early studies even prior to whole-genome expression profiling (Akutsu, Miyano, & Kuhara, 1999; Duggan, Bittner, Chen, Meltzer, & Trent, 1999; Liang, Fuhrman, & Somogyi, 1998). Driven by new datasets, a raft of recent computational methods now predict genetic perturbation results *in silico*, with applications in embryology, stem cell reprogramming, and drug discovery (Amrute et al., 2022; Burdziak et al., 2023; Cui, Wang, Maan, & Wang, 2023; Hyttinen, Eberhardt, & Hoyer, 2012; Jiang et al., 2023; Kamimoto et al., 2023; Osorio et al., 2022; Qiu et al., 2022; Roohani, Huang, & Leskovec, 2022; Theodoris et al., 2023; Tran, Yang, Yang, & Ormerod, 2022; Wang et al., 2022; Yeo, Saksena, & Gifford, 2021). These expression forecasting methods aim to serve as a new type of general-purpose screening tool. Compared to Perturb-seq and similar assays, computer simulations are cheaper, less labor-intensive, and easier to apply to less accessible cell types. Example applications frequently include ranking or nomination of hundreds of genetic perturbations, sometimes in cell types where very few genetic perturbations have been conducted (Burdziak et al., 2023; Jiang et al., 2023; Kamimoto et al., 2023; Roohani et al., 2022; Theodoris et al., 2023; Wang et al., 2022; Yeo et al., 2021). This could enable more efficient research in a variety of areas; examples might include determining genetic dependencies of cell fate transitions in cancer (Baca et al., 2021; Reddy et al., 2021) or using reprogramming to reverse molecular indicators of aging (Lee et al., 2021).

The utility of these approaches hinges on their accuracy. A number of rationales are provided for the feasibility of expression forecasting under novel genetic perturbations, including mathematical identifiability guarantees (Yeo et al., 2021), use of prior functional annotations (Roohani et al., 2022), support for TF-to-target binding via ChIP or motif analysis (Burdziak et al., 2023; Kamimoto et al., 2023; Tran et al., 2022; Wang et al., 2022), and empirical tests against real perturbation outcomes (Cui et al., 2023; Jiang et al., 2023; Kamimoto, Hoffmann, & Morris, 2020; Qiu et al., 2022; Roohani et al., 2022; Theodoris et al., 2023; Wang et al., 2022; Yeo et al., 2021). However, most expression forecasting methods are very new, and most have been tested on only a small number of perturbations, with few direct comparisons across methods (Cui et al., 2023; Jiang et al., 2023; Lopez, Hütter, Pritchard, & Regev, 2022; Roohani et al., 2022; Theodoris et al., 2023; Tran et al., 2022; Wang et al., 2022). Expression forecasting benchmarks on mammalian data have often been conducted by the authors of competing prediction tools, raising the possibility of over-optimistic results due to unknown experimenter degrees of freedom (Boulesteix, Lauer, & Eugster, 2013; Nießl, Herrmann, Wiedemann, Casalicchio, & Boulesteix, 2022). Modern datasets offer the opportunity to test detailed predictions about perturbation outcomes at unprecedented scale (Dixit et al., 2016; Nakatake et al., 2020; Replogle et al., 2020, 2022; Subramanian et al., 2017) (Adamson et al., 2016; Norman et al., 2019). Our work complements existing reports by providing neutral, systematic prediction benchmarks on a diverse collection of genetically perturbed human cell types.

Methods for expression forecasting differ in terms of expected output. Some methods provide only cell-type classifications or low-dimensional embeddings (Cui et al., 2023; Kamimoto et al.,

2023; Theodoris et al., 2023; Yeo et al., 2021), while others provide gene-specific predictions of expression, velocity, or fold-change (Burdziak et al., 2023; Qiu et al., 2022; Roohani et al., 2022; Wang et al., 2022). To build flexible and extensible benchmarking infrastructure, we have collected datasets with transcriptome-wide profiles of many genetic perturbations. These could be used to benchmark gene-by-gene fold changes or derived cell state labels and low-dimensional embeddings, though we initially focus on gene-by-gene fold changes to extract the maximum amount of information from the data.

Expression forecasting methods also vary widely in terms of implementation and expected input. Certain methods use prior drafts of causal network structure (CellOracle, Dictys, ScanBMA, scREMOTE, scKINETICS, D-SPIN); prior knowledge of gene-gene functional relatedness (GEARS); or pre-training on millions of cells (scGPT, GeneFormer) (Burdziak et al., 2023; Cui et al., 2023; Jiang et al., 2023; Kamimoto et al., 2023; Lopez, Hütter, et al., 2022; Pemberton-Ross, Pachkov, & van Nimwegen, 2015; Qiu et al., 2022; Roohani et al., 2022; Theodoris et al., 2023; Wang et al., 2022; Yeo et al., 2021; Young, Raftery, & Yeung, 2014). Regarding analytical choices, key distinctive features include different regression methods (DCD-FG, CellOracle, Dynamo), use of low-rank structure (DCD-FG, ARMADA, D-SPIN); special handling of complex time-series (PRESCIENT); and modeling of transcription rates in addition to transcript levels (Dynamo, scKINETICS). For many expression forecasting methods, a key design element is a sparse, interpretable gene regulatory network (GRN) inferred from a combination of motif analysis and gene expression data. We develop a grammar of GRN inference that synthesizes certain key distinguishing properties of the methods inspiring this study. We provide software to enable easy combination of key methodological features, which we then deploy for benchmarking.

Results

Large-scale perturbation datasets display a high rate of success in targeting individual genes, modest correlation across replicates, and mostly small global effects

There are many published instances of medium- or large-scale genetic perturbation data, but to make these readily accessible for expression forecasting benchmarks, they require quality control and uniform annotation and formatting. We summarize the properties of the data used in this study (Table 1) and provide due diligence regarding data exploration and perturbation effects (Fig S1).

Table 1

Name in this project	Technology	DOI	Genes perturbed	Gene selection	Perturbation type	Take-down day	Cell type
Nakatake	Overexpression + bulk RNA	10.1016/j.celrep.2020.107655	714	Curated	Over-expression	2	PSC
Freimer	TF knockdown + bulk RNA	10.1038/s41588-022-01106-y	24	Regulators of IL2RA, IL-2 and CTLA4	Knockout	5	CD4+ T cells
Replogle	CRISPRa + 10X 3' scRNA	10.1038/s41587-020-0470-y	63	Manual	Over-expression	8	K562
Replogle2	CRISPRi + 10X 3' scRNA	10.1016/j.cell.2022.05.013	9867	All expressed genes	Knockdown	8	K562
Replogle3	CRISPRi + 10X 3' scRNA	10.1016/j.cell.2022.05.013	2285	All essential genes	Knockdown	6	K562
Replogle4	CRISPRi + 10X 3' scRNA	10.1016/j.cell.2022.05.013	2285	All essential genes	Knockdown	7	RPE1
Frangieh	CRISPR + 10X 3' scRNA	10.1038/s41588-021-00779-1	248	CRISPR screen for survival	Knockout	14	Primary melanoma
Adamson	CRISPRi + 10X 3' scRNA	10.1016/j.cell.2016.11.048	87	UPR-related genes	Knockdown	7	K562
Dixit	CRISPR + 10X 3' scRNA	10.1016/j.cell.2016.11.038	24	(Amit et al., 2009)	Knockout	7	mouse BMDC

In each dataset, we investigated both global and targeted-locus effect size (Fig S1). Where possible, we display the Spearman correlation in log fold change between replicates, but in cases lacking biological replicates, we display correlation between technical replicates (e.g. different guide RNA's) or we show the number of differentially expressed genes. The datasets contain a minority of perturbations with very large effects, and these larger effects tend to have higher correlation between replicates. For each overexpression or knockdown experiment, targeted genes' expression mostly increases or decreases as expected, with the lowest rate being the Nakatake and Replogle overexpression datasets: in both, about 92% of overexpressed transcripts increase as expected. In knockout experiments, cells with a nonsense mutation or a truncated sequence can in principle express high levels of detectable mRNA while experiencing complete loss of function of the encoded protein, but we still observe that target gene RNA levels frequently decrease. In knockdown and overexpression experiments, we removed samples where the targeted transcript does not decrease or increase as expected. In knockout experiments, we retained all samples except those flagged for quality issues by the original authors. For certain datasets used by DCD-FG and GEARS, we use preprocessing identical to prior evaluations. A full accounting of our preprocessing pipeline is given in Methods.

In most of these datasets, genes to perturb are not selected at random, but rather are known to be important for survival or regulation (Table 1). One exception is the Replogle2 genome-wide Perturb-seq, in which all genes are knocked down. To understand how non-random perturbation

selection affects results, our collection separately labels subsets of the Replogle2 genome-wide Perturb-seq that include 1) only TF perturbations or 2) only large effects.

Figure S1

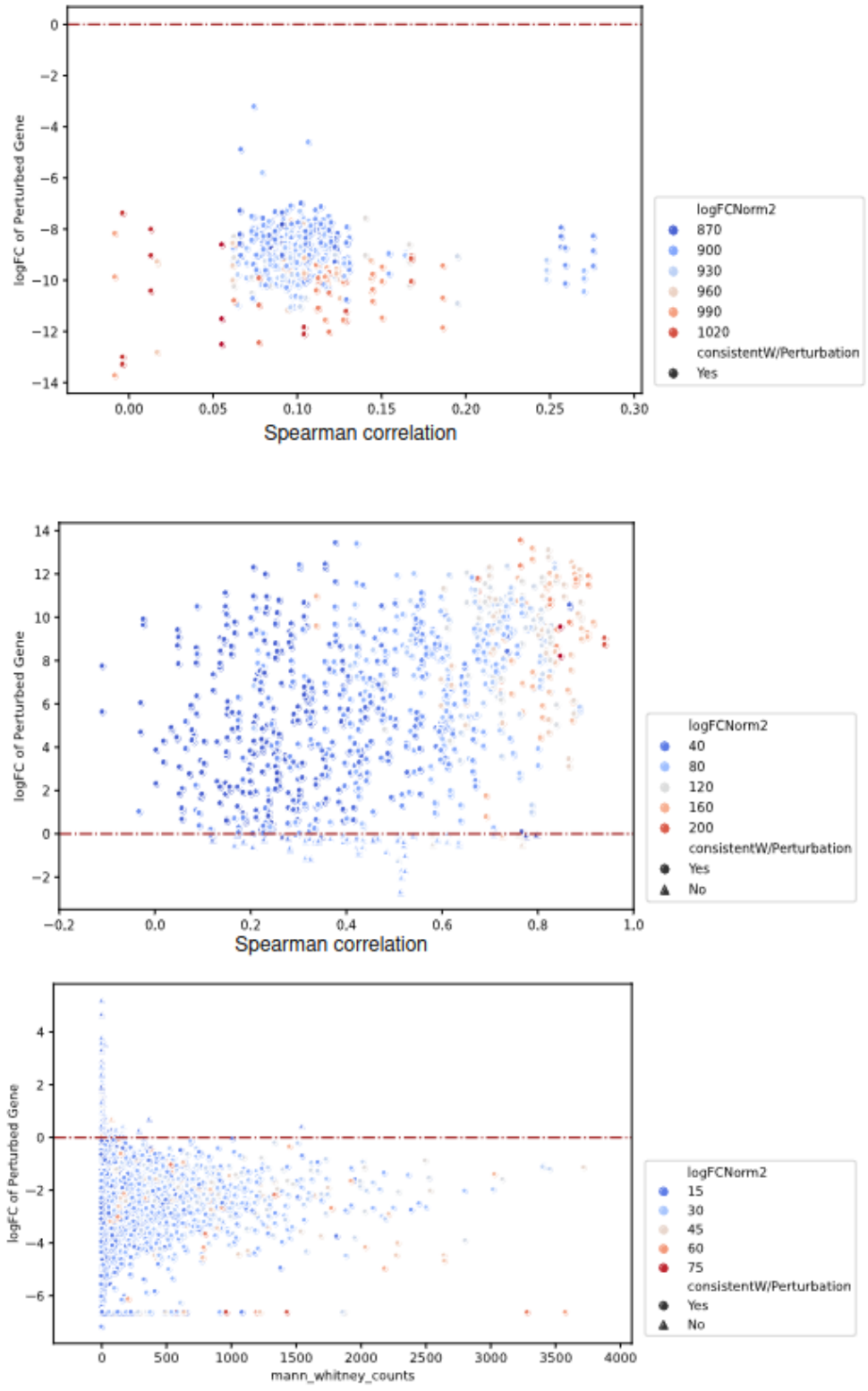
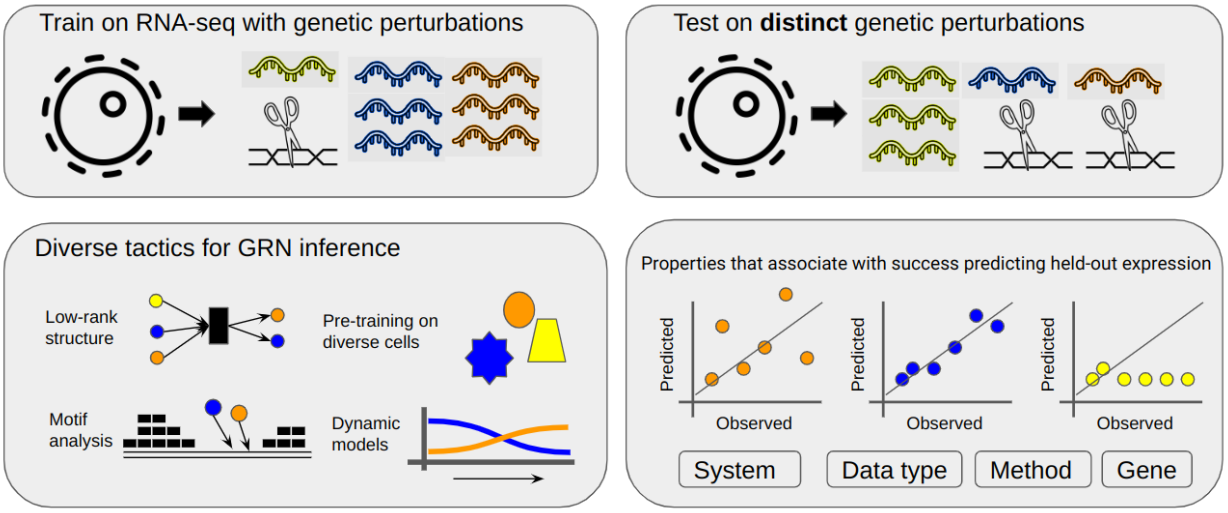

logFC of Perturbed Gene
Spearman correlation
logFCNorm2
870
900
930
960
990
1020
consistentW/Perturbation
Yes
logFC of Perturbed Gene
Spearman correlation
logFCNorm2
40
80
120
160
200
consistentW/Perturbation
Yes
No
logFC of Perturbed Gene
mann_whitney_counts
logFCNorm2
15
30
45
60
75
consistentW/Perturbation
Yes
No

Fig S1. Quality analysis of large perturbation datasets.

A. Perturbation effect size on the knocked-out gene (Y-axis) and the whole transcriptome (color scale), with Spearman correlation of fold change between different guide RNA's (X-axis). Based on the interferon-gamma treated subset of the Frangieh melanoma data.
B. Perturbation effect size on the overexpressed gene (Y-axis) and the whole transcriptome (color scale), with Spearman correlation of fold change between replicates (X-axis). Based on the Nakatake PSC data.
C. Perturbation effect size on the knocked-down gene (Y-axis) and the whole transcriptome (color scale, X-axis). The X-axis quantity "mann_whitney_counts" refers to the number of differentially expressed genes (Replogle et al., 2022). Based on the 'Replogle2' K562 data.

Figure 2


Train on RNA-seq with genetic perturbations
Test on **distinct** genetic perturbations
Diverse tactics for GRN inference
Low-rank structure
Pre-training on diverse cells
Motif analysis
Dynamic models
Properties that associate with success predicting held-out expression
Predicted
Observed
Predicted
Observed
Predicted
Observed
System
Data type
Method
Gene

Regression-based analyses do not out-perform uninformed baselines

Our initial experiments characterize performance of simple regression-based expression forecasting schemes. To explore the selection of regression methods and input features, we conducted a series of tests based on prediction of log fold changes (Fig. 2). A key feature of these evaluations is complete lack of overlap between the perturbations in the training set and the perturbations in the test set. No gene is perturbed in both the training and the test set, emphasizing the role of these methods for *in silico* screening rather than efficient estimation of effect sizes once perturbations are observed. We created a configurable benchmarking framework that allows users to easily specify an evaluation dataset and a regression method. For each gene, the specified regression method is trained using as labels the expression of each target gene and using as features the expression of all TF's (except the gene itself). To predict perturbation outcomes, we begin with the average expression of all negative controls. The corresponding transcription factor is then set to 0 (for knockout experiments) or to its observed value after intervention (for knockdown or overexpression experiments). From the resulting perturbed features, regression models are used to make predictions. This experimental design would be expected to emphasize direct effects rather than total effects, and results might differ if training focused on total effects or if predictions were iterated until steady state (Wang et al., 2022).

We began with a train-test split of each dataset, putting 50% of the genetic perturbations in the test set. We tested nine supervised learning methods available in scikit-learn (Pedregosa et al., 2011). To our surprise, all performed decisively worse than the mean or median of the training data on all datasets we tested (Fig. 3a right) across many evaluation metrics (Fig. S2), with two exceptions being LASSO regression on Nakatake when evaluating MSE of logFC predictions on the top 20 genes and kernel ridge regression on Freimer when evaluating MSE of logFC predictions on the top 20 genes (Fig. S2). The top 20 genes are selected on observed absolute log fold change, mirroring previous evaluations that aimed to avoid advantaging trivial methods (Roohani et al., 2022).

Suspecting overfitting or lack of causal idenfitication, we sought to reduce the sample complexity of this task. Many methods restrict causal relationships by predicting a target gene's expression using only TF's with nearby motif occurrences in promoters or enhancers (Kamimoto et al., 2023; Pemberton-Ross et al., 2015; Wang et al., 2022; Young et al., 2014), yielding regression problems with fewer predictors and therefore lower-variance estimates. We collected over 10 previously published networks constructed by coexpression analysis, Bayesian edge prediction, and motif analysis of candidate cis-regulatory elements (Table 4). We processed them into a uniform storage format and incorporated them into our benchmarking software. The software allows users to easily train diverse supervised ML methods using expression of previously nominated neighboring genes as the input features.

Separately using each of several networks for feature selection, we trained ridge regression models with TF expression as features and target gene expression as labels. We predicted expression after held-out interventions and evaluated performance. As controls, we included a dense network (all features are used for each target) and an empty network (no features are

used). The CellNet, GTEx, and FANTOM4 networks out-perform the empty network in some cases (Fig. 3a right). However, the empty network in this case corresponds to the mean of the training data, whereas the best-performing regression method is typically the median, and no network structures out-performed the median. When using alternative evaluation metrics, some networks out-perform simple baselines, notably the CellNet network based on HuGene array data evaluated on a subset of Replogle2 with only TF's perturbed. Overall, the median and the empty network still perform best (Fig. S2).

To determine whether evaluation on held-out perturbations can ever yield meaningful results, we simulated data from an autoregressive model based on several of the known network structures. The network used to generate the data performs yields lowest MAE in four of the five evaluations and beats trivial baselines in all cases, showing that held-out perturbations can provide meaningful comparison of alternative methods (Fig. 3d).

Even if expression forecasting methods do not outperform baseline predictors in terms of transcriptome-wide mean absolute error, they may work for certain target genes or perturbations. Taking advantage of a genome-wide perturb-seq (labeled Replogle2 in Table 1), we measured performance separately as if only TF's had been perturbed or only genes with large effects had been perturbed. Trivial baselines performed best.

We also measured mean absolute error for each target gene separately and for each perturbed gene separately. We stratified the results by many features, including probability of being loss-of-function intolerant (pLI), (Lek et al., 2016); in-degree or out-degree from published networks; mean, variance, or dispersion rank in training data; and expression characteristics in the test data. We analyzed stratified evaluation results for tests of different sklearn methods on Nakatake, Replogle2, and Frangieh. On all three, stratification of target genes produced no groups of genes where informative methods beat baselines (data not shown), but stratification of perturbed genes sometimes produced groups of perturbations where informative methods beat baselines. In the Nakatake data, informative methods beat baselines only in the following cases: perturbations of genes with high mean expression and low dispersion (Fig. 3c); perturbations with low effect sizes on either the targeted gene and the whole transcriptome (Fig. 3d); perturbations with out-degree 0 in a pruned version of the ANANSE networks (data not shown); and perturbations of genes with high in-degree in a protein-protein interaction network (data not shown). In each of these cases, the highest performer was ExtraTrees (Geurts, Ernst, & Wehenkel, 2006). In the Replogle2 dataset, likewise results were largely negative, but several methods beat baselines on perturbations of genes with low mean and low variance or with a number of specific degree characteristics (data not shown). In the Frangieh dataset, the only victories over baseline were on perturbations of genes with medium to high degree in regulatorynetworks.org human and mouse networks (data not shown); stratification by pLI, expression characteristics, or degree characteristics in other networks was not useful. No feature enriched for perturbations that are predicted better than random across multiple datasets.

Figure 3

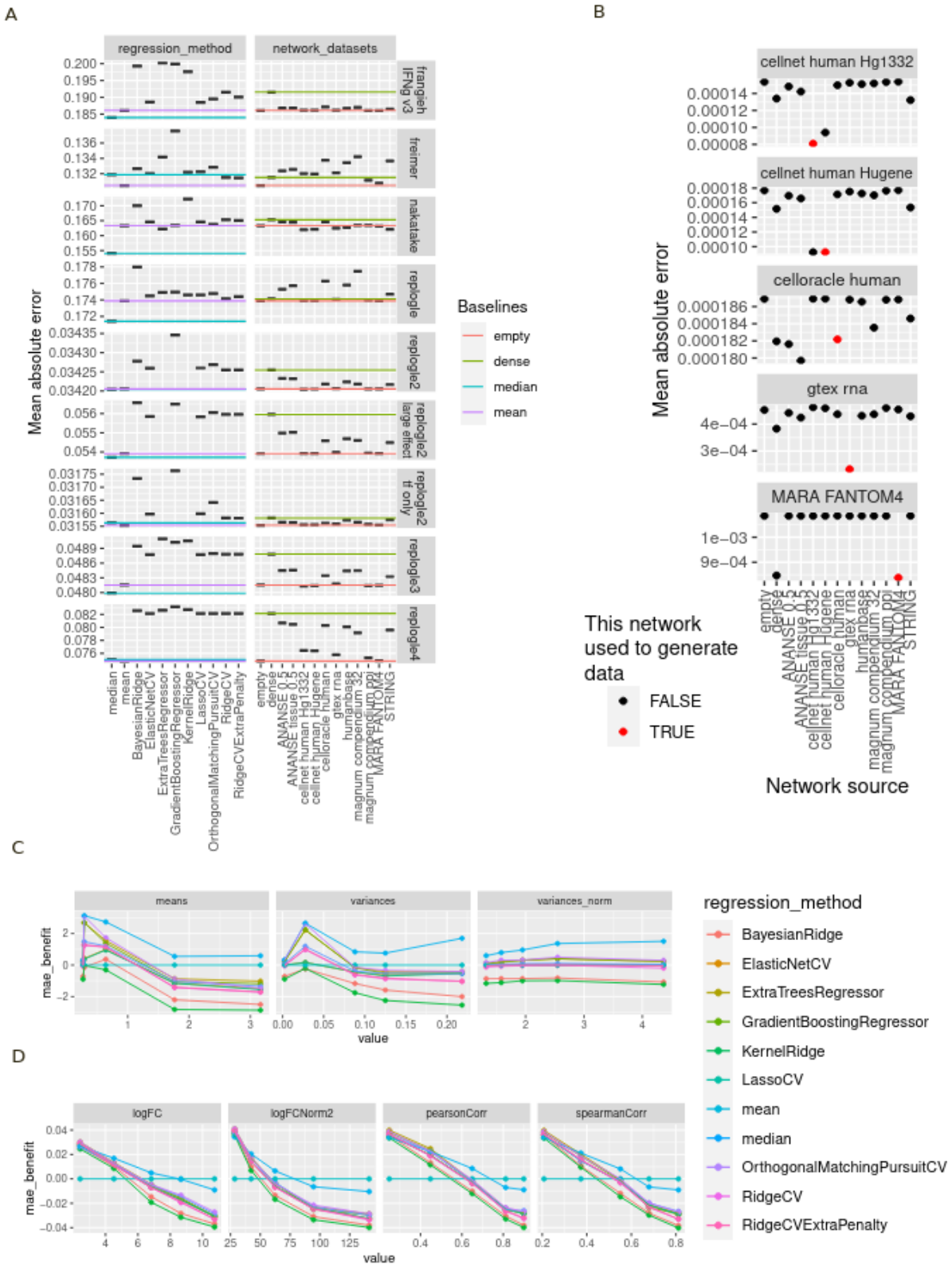

A
regression_method
network_datasets
Mean absolute error
frangieh IFNg v3
freimer
nakatake
replogle
replogle2
replogle2 large effect
replogle2 tf only
replogle3
replogle4
Baselines
empty
dense
median
mean
median
mean
BayesianRidge
ElasticNetCV
ExtraTreesRegressor
GradientBoostingRegressor
KernelRidge
LassoCV
OrthogonalMatchingPursuitCV
RidgeCV
RidgeCVExtraPenalty
empty
dense
ANANSE 0.5
ANANSE tissue 0.5
cellnet human Hg1332
cellnet human Hugene
celloracle human
gtex rna
humanbase
magnum compendium 32
magnum compendium ppi
MARA FANTOM4
STRING
B
cellnet human Hg1332
cellnet human Hugene
celloracle human
gtex rna
MARA FANTOM4
Mean absolute error
This network used to generate data
FALSE
TRUE
Network source
C
means
variances
variances_norm
mae_benefit
value
regression_method
BayesianRidge
ElasticNetCV
ExtraTreesRegressor
GradientBoostingRegressor
KernelRidge
LassoCV
mean
median
OrthogonalMatchingPursuitCV
RidgeCV
RidgeCVExtraPenalty
D
logFC
logFCNorm2
pearsonCorr
spearmanCorr
mae_benefit
value

Figure 3 Evaluation of supervised ML and feature selection methods against simple baseline predictors.

a. Mean absolute error (Y axis) for the indicated forecasting method or network structure (X axis) when predicting log fold change under held-out perturbations. There is no replication. The data split is identical between the regression method experiments and the network experiments, so that values are directly comparable between left and right columns. ExtraTrees and KernelRidge are missing from certain panels due to memory requirements exceeding 64GB.
b. As in panel A, but data are simulated from an autoregressive model using the indicated network structure.
c. This evaluation is based on only the Nakatake data. Improvement versus the mean of the training data in the MAE (Y axis) is stratified by characteristics of each target gene in the training data (X axis). “Mean” and “variance” are the mean and variance of the log-transformed expression. “Variances_norm” is the measure of dispersion computed by scanpy during highly variable gene selection.
d. As in panel C, but performance is stratified by characteristics of each perturbation measured in the test data (X axis). “LogFC” is the log fold change in the gene that was overexpressed. “LogFCnorm2” is the sum of squares of the fold changes of all transcripts. “pearsonCorr” and “spearmanCorr” measure the correlation between log fold changes for all transcripts in different replicates of each perturbation. Each dot shows the median X and Y values in a fixed-size bin imposed on the X axis.

Figure S2

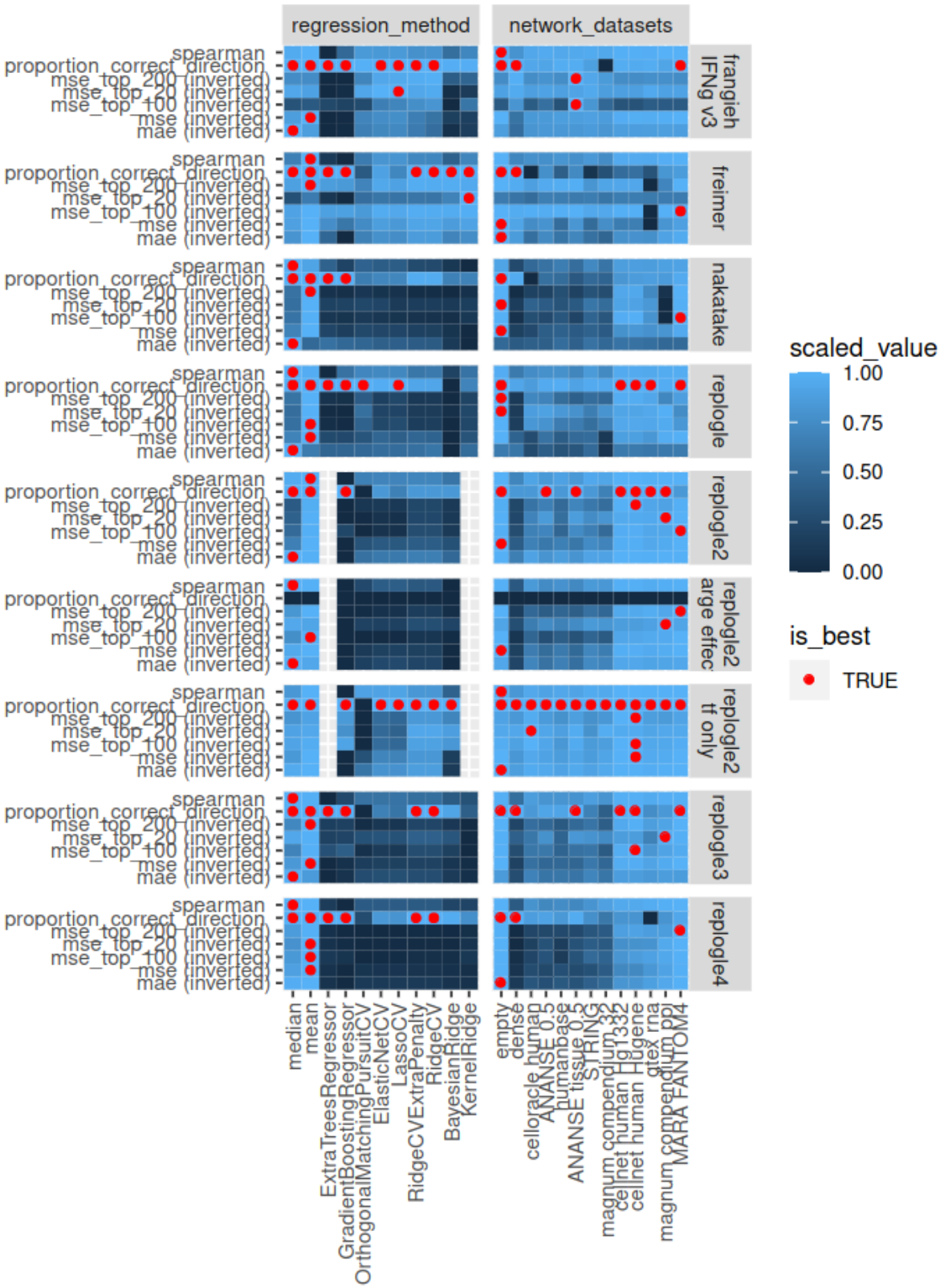

regression_method
network_datasets
spearman
proportion_correct_direction
mse_top_200 (inverted)
mse_top_20 (inverted)
mse_top_100 (inverted)
mse (inverted)
mae (inverted)
frangieh IFNg v3
freimer
nakatake
replogle
replogle2
replogle2 large effect
replogle2 tf only
replogle3
replogle4
median
mean
ExtraTreesRegressor
GradientBoostingRegressor
OrthogonalMatchingPursuitCV
ElasticNetCV
LassoCV
RidgeCVExtraPenalty
RidgeCV
BayesianRidge
KernelRidge
empty
dense
celloracle human
ANANSE 0.5
humanbase
ANANSE tissue 0.5
STRING
magnum compendium 32
celnet human Hg132
celnet human Hugene
gtex rna
magnum compendium ppi
MARA FANTOM4
scaled_value
1.00
0.75
0.50
0.25
0.00
is_best
TRUE

Figure S2. Related to figure 3. Alternative performance metrics for prediction of held-out perturbations. All metrics are linearly transformed so the best performance is 1 and the worst is 0; metrics are marked “inverted” if they would ordinarily have lower numbers for better performance. **Spearman**, spearman correlation between observed and predicted log fold change; **Proportion correct direction**, fraction of target genes with the correct direction of change; **mse_top_n**, mean squared error on the top n genes ranked by observed absolute log fold changes; **mse,** mean squared error on all genes; **mae**, mean absolute error on all genes. The data split and the scaling of each metric is identical between the regression method experiments and the network experiments, so that values are directly comparable between left and right columns. ExtraTrees and KernelRidge are missing from certain panels due to memory requirements exceeding 64GB.

Table 4

Internal name	Description	Data used	Species	DOI
CellNet	CLR analysis of microarray data from GEO	Expression microarrays	human, mouse	10.1016%2Fj.cell.2014.07.020
encode-nets_human	TF-ChIP data from ENCODE	TF ChIP	human	10.1038/nature11245
humanbase	Bayesian classifier based on GO term co-occurrence.	Variety of genomic data	human	10.1038/ng.3259
chea	TF ChIP data (miscellaneous)	ChIP-X	human, mouse	10.1093/bioinformatics/btq466
csnets	Motif analysis + RNA integration	Dnase, RNA	human	10.1186/s12864-018-5277-6
MAGNUM	Motif analysis of FANTOM5 CAGE data.	CAGE	human	10.1038/nmeth.3799
GNAT	Graphical LASSO on gTEX data with tree-structured random effects	RNA-seq	human	10.1371/journal.pcbi.1004220
celloracle_human	Motif analysis of promoters	None	human	10.1101/2020.02.17.947416
MARA_FANTOM4	MARA on FANTOM4 THP-1 data	CAGE	human	10.1038/ng.375
ANANSE	Motif analysis of enhancers	ATAC and ChIP	human	10.1093/nar/gkab598

Table 4. Published networks that are expected to be enriched for human transcriptional regulatory relationships.

Diverse published methods do not out-perform simple baselines

Given the preceding largely negative findings, we sought to test expression forecasting methods that had already been empirically validated, extending initial promising experiments to a broader collection of datasets. Based on the type of training data we have available, we selected three methods: DCD-FG, GEARS, and GeneFormer (Lopez, Hütter, et al., 2022; Roohani et al., 2022; Theodoris et al., 2023).

Using code published by the DCD-FG authors, we repeated the benchmarks of DCD-FG on the gamma interferon-treated subset of the Frangieh melanoma data. The relative performance of DCD-FG and related methods NO-TEARS-LR and NO-TEARS is consistent with Lopez et al. (Fig. 5a). In addition to the methods implemented in DCD-FG, we included an IID Gaussian baseline, and we fit either a diagonal or a full covariance matrix. Surprisingly, the IID baseline outperformed all causal inference methods when using a full covariance matrix (Fig. 5a).

The relatively strong performance of this simple, non-interventional baseline motivated us to study the DCD-FG benchmarks in closer detail. When predicting a target gene's expression, the DCD-FG benchmarks allow each method to use held-out expression of genes that are causally upstream according to the learned model (Romain Lopez, personal communication). This is a natural choice due to the mathematical formulation of DCD-FG, but it does not fit the use-case motivating our study, in which held-out perturbations are completely unobserved. Thus, we modified our benchmarking framework to include DCD-FG and to impose explicit control over the use of held-out gene expression during prediction. DCD-FG benefits substantially from using held-out expression of upstream genes as opposed to control expression (Fig. 5b). Based on the criteria used in our other benchmarks, which do not reveal any held-out expression during prediction, DCD-FG outperforms NOTEARS-LR significantly ($p<2.2\times10^{-16}$, paired t-test) and the mean non-significantly ($p=0.91$ paired t-test), but DCD-FG does not out-perform the median of the training data (Fig. 5b top left).

Since the DCD-FG model does not include measurement error and the melanoma data contain many perturbational regimes with only 1 cell measured (66% of regimes), measurement error may substantially affect results. We repeated the Frangieh benchmarks after removing perturbational regimes with fewer than 50 cells and averaging expression within each perturbational regime. This preprocessing eliminated combinations of knockouts and focused the evaluation on entirely perturbations of genes that are not perturbed in the training data, rather than novel combinations of genes that may be perturbed individually in the training data. DCD-FG performed worse than either the mean or the median of the training data, even if held-out expression was used (Fig. 5b top middle).

On the Nakatake data, which are bulk transcriptome data, DCD-FG out-performed the mean and median only when using held-out expression of upstream genes (Fig. 5b top right). On other datasets, only control expression was tested as the starting point, and DCD-FG never out-performed simple baselines (Fig. 5b bottom rows).

Figure 5

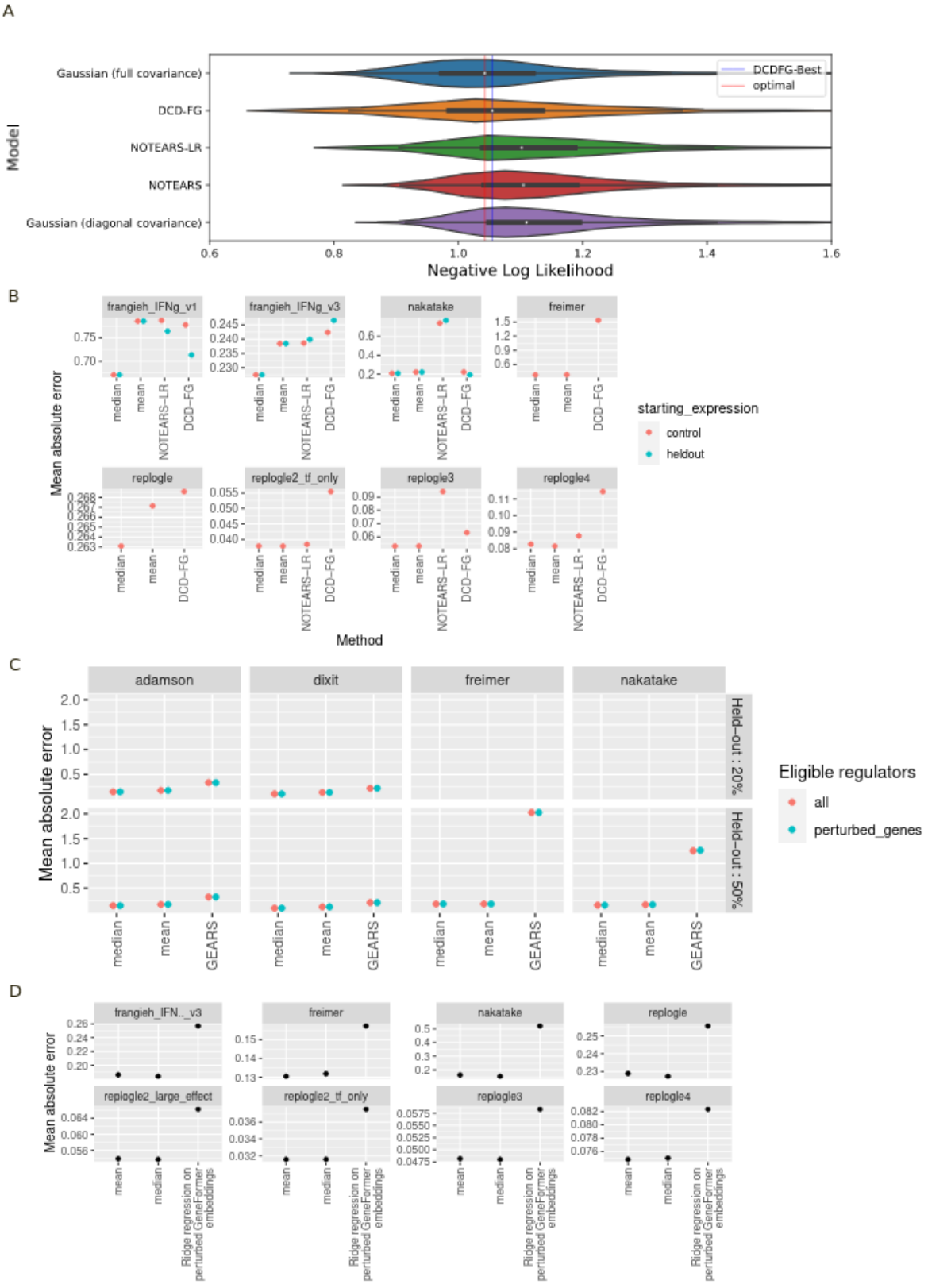

A
Gaussian (full covariance)
DCD-FG
NOTEARS-LR
NOTEARS
Gaussian (diagonal covariance)
Model
DCDFG-Best
optimal
0.6
0.8
1.0
1.2
1.4
1.6
Negative Log Likelihood
B
frangieh_IFNg_v1
frangieh_IFNg_v3
nakatake
freimer
replogle
replogle2_tf_only
replogle3
replogle4
Mean absolute error
median
mean
NOTEARS-LR
DCD-FG
Method
starting_expression
control
heldout
C
adamson
dixit
freimer
nakatake
Held-out : 20%
Held-out : 50%
Mean absolute error
median
mean
GEARS
Eligible regulators
all
perturbed_genes
D
frangieh_IFN..._v3
freimer
nakatake
replogle
replogle2_large_effect
replogle2_tf_only
replogle3
replogle4
Mean absolute error
mean
median
Ridge regression on perturbed GeneFormer embeddings

Figure 5: Evaluations of published methods.

a. DCD-FG evaluation. Negative log likelihood on held-out perturbations in the Frangieh melanoma data, performed using the original code for DCD-FG with IID Gaussian baselines added.
b. DCD-FG evaluation using GGRN. **Top row:** DCD-FG benchmarks repeated in our benchmarking framework using as a initial state either the **control** or **held-out** expression of upstream genes, and using two different methods of preprocessing the Frangieh melanoma data or using 1000 most variable genes in the Nakatake PSC data. Briefly, v1 of the Frangieh melanoma data follows Lopez et al exactly, whereas v3 includes only single-gene perturbations and averages the data within each combination of guide RNAs. **Bottom rows:** DCD-FG benchmarks on other datasets using only **control** expression as the initial state.
c. GEARS evaluation. Mean absolute error for GEARS versus simple baselines when predicting held-out perturbations on different data splits (row) of different datasets (column). Color shows the genes used to construct GO graph: either all genes, or the union of all perturbed genes across the training and test sets.
d. GeneFormer evaluation. Improvement in mean absolute error for ridge regression versus simple baselines when predicting held-out perturbations using GeneFormer cell embeddings for post-perturbation feature extraction on different datasets (facet).

GEARS is another expression forecasting method that has been successfully tested on held-out genetic perturbations or held-out combinations of genetic perturbations (Roohani et al., 2022). Compared to methods mentioned so far, GEARS uses a different principle to generalize to genes not perturbed in the training data: a graph neural network that enforces similar outcomes for genes with similar membership in Gene Ontology terms. GEARS has shown promising results when predicting fold change due to held-out perturbations.

We trained GEARS on the 50% of the perturbations in the Nakatake data, then tested on the remaining 50%. GEARS yielded a worse MAE than simple baselines (Fig. 5 c). However, GEARS was developed for single-cell data, whereas the Nakatake data are in bulk. For an evaluation on single-cell data, we tested a 50-50 data split on two Perturb-seq datasets (Adamson et al., 2016; Dixit et al., 2016), using preprocessing identical to the GEARS paper. GEARS yielded a worse MAE than simple baselines. We repeated with an 80%-20% train-test split and we tested GEARS on all datasets used above, omitting only tests with runtime errors, but no results were better than baseline (Fig. 5 c).

GeneFormer was recently claimed to predict changes in cardiac muscle state in response to genetic perturbation, even without being trained on any genetic perturbations (Theodoris et al., 2023). A promising component of GeneFormer's strategy, not included in any methods tested above, is pretraining of a Transformer-based language model on a corpus of 30 million cells. GeneFormer is intended to predict low-dimensional summaries of cell state, not per-gene fold changes, but we reasoned that GeneFormer embeddings may still include uniquely useful information for expression forecasting.

We integrated GeneFormer into our regression-based forecasting framework as a feature extraction method. We constructed post-perturbation features by re-ranking the tokens representing the perturbed genes and extracting latent cell embeddings, which is the strategy used for GeneFormer's *in silico* treatment analysis. Ridge regression models were trained to "decode" the extracted features into post-perturbation log-scale gene expression. However, this strategy did not out-perform the mean or median of the training data (Fig. 5d). We note that this is not the only possible way of re-purposing GeneFormer for expression forecasting.

A grammar of gene regulatory network (GRN) inference synthesizes diverse methods within an extensible benchmarking framework

Existing expression forecasting methods represent individual points in a space of all possible methods. Methods development can be viewed as a search over these points, but existing software does not facilitate such a search. To enable flexible combination of GRN inference features, we developed a conceptual framework called GGRN (for "Grammar of Gene Regulatory Networks"), as well as a Python implementation. This framework characterizes expression forecasting methods according to their handling of time-scales, supervised learning problems, low-dimensional projection, known interactions among genes, and sharing of information across cell types. A full exposition is provided in Supplemental File 1.

Briefly, the most fundamental attribute distinguishing GRN methods is how they construct a training dataset. Let $X_i(P, T)$ be RNA abundance in sample i, at time T. Suppose genes in the set P were overexpressed continuously starting at T=0. A natural approach is to fit F such that $X_i(P, T + dT) \approx F(X_i(P, T))$. To estimate F via supervised machine learning, it is necessary to create a training set with matched pairs separated by a known span of time. The most common scheme is to assume that samples have reached a steady state and to fit $X_i(P, T) \approx F(X_i(P, T))$; this is done by LLC, CellOracle, Dictys, and DCD-FG (Hyttinen et al., 2012; Kamimoto et al., 2023; Lopez, Hütter, et al., 2022; Wang et al., 2022). If time-series data are available, an alternative is to match each sample with a counterpart at a previous time-step, which is done on bulk transcriptome data by ScanBMA and ARMADA. With single-cell RNA data, this type of matching is harder, requiring lineage tracing predictions. In one method, PRESCIENT, lineage tracing is solved by optimal transport matching between forward simulation results and observed cells at the next time-point. Another way to obtain paired measurements across two timepoints is RNA velocity, which is used by Dynamo. Finally, GEARS and scGPT match up perturbed profiles with randomly selected control cells, fitting models directly to fold changes at the takedown time-point (Cui et al., 2023; Roohani et al., 2022).

Given a training set of matched pairs separated by a known span of time, several other decisions remain. The transition function F can be learned via a variety of regression methods, including ridge regression variants (CellOracle, Dynamo, ARMADA), linear stochastic differential equations (Dictys), or various neural networks (GEARS, PRESCIENT, DCD-FG). A variety of time-points can be offered as the final forecast, including one (DCD-FG), three (CellOracle), the final steady state (Dictys), or a full trajectory (PRESCIENT, Dynamo). Feature construction can be based on motif analysis in a variety of ways, such as selection of relevant TF mRNA's (CellOracle, Dictys), modeling of latent TF activity (ARMADA), or embedding via a pretrained foundation model (GeneFormer). Perturbations can be modeled as persistent (CellOracle, Dynamo) or transient (PRESCIENT). Low dimensional representations can be used mainly for projection of output (Dynamo, CellOracle); or prior to all modeling (PRESCIENT, DSPIN); or in ways that are tightly integrated with mechanisms represented by F (ARMADA, DCD-FG, GEARS). In the supplementary information, we further discuss how to systematically combine these different attributes.

This flexible, systematic framing of expression forecasting facilitates fair comparison of diverse methods. The accompanying software is readily extensible, interfacing with any expression forecasting method via Docker. We hope this work will provide the community with easy access to thorough expression forecasting benchmarks.

Discussion

Prior benchmarks of gene regulatory network inference in mammals have mostly focused on structure recovery (Chevalley, Roohani, Mehrjou, Leskovec, & Schwab, 2022; Lasri, Shahrezaei, & Sturrock, 2022; Marbach et al., 2012; McCalla et al., 2023; Pratapa, Jalihal, Law, Bharadwaj, & Murali, 2020; Saint-Antoine & Singh, 2023), which is particularly challenging (Badia-I-Mompel et al., 2023). Some structure benchmarks use protein-protein interactions,

which does not establish a causal effect on transcription. Others use synthetic data, which in general can favor methods that do not succeed on real data (Pawel, Kook, & Reeve, 2023) and in network inference specifically may lack important aspects of real data such as measurement error, high dimensionality, and unobserved causal factors. Some use curated gold standards with few or no negative examples, or ChIP-based gold standards that measure binding regardless of the effect on transcription, or perturbation-based gold standards that do not distinguish direct from indirect effects. Furthermore, many recent methods avoid structure inference entirely or view it as a byproduct, with the primary goal instead being perturbation outcomes (Cui et al., 2023; Hyttinen et al., 2012; Kamimoto et al., 2023; Osorio et al., 2022; Qiu et al., 2022; Roohani et al., 2022; Theodoris et al., 2023; Wang et al., 2022; Yeo et al., 2021).

Benchmarking of predicted expression sidesteps these difficulties. It has been used to demonstrate detailed understanding of an archaeon (Bonneau et al., 2006, 2007) and to compare causal inference methods in yeast (Maathuis, Colombo, Kalisch, & Bühlmann, 2010; Meinshausen et al., 2016). Our main contribution is to extend this promising benchmarking approach to human data.

Expression forecasting on human data also enables head-to-head comparison of diverse sources of prior knowledge. In particular, we test numerous methods using draft TF-target networks, as well as GEARS, which uses prior knowledge of functional relatedness, and GeneFormer, which is pretrained on a large collection of scRNA data. Most existing benchmarks focus on methods alone. In particular, CausalBench (Chevalley, Roohani, et al., 2022) and the affiliated GSK challenge (Chevalley, Mehrjou, Schwab, Notin, & Roohani, 2022) explicitly disallow use of prior knowledge, so although we use some of the same evaluation data as CausalBench, our study differs in scope.

Our results are largely negative, with very simple baselines typically performing at least as well as any of the benchmarked methods. This is unexpected given prior results. To improve future benchmarks and methods development, it is important to understand the drivers of each discrepancy. Where possible, we tested specific technical choices, revealing that favorable benchmarks for DCD-FG require use of test-set expression of putatively upstream genes to predict putatively downstream genes. Similar tasks have been used for GRN evaluation (Kamal et al., 2023; Lopez, Tagasovska, et al., 2022; Miraldi et al., 2019), but predictions based on partial knowledge of post-perturbation expression cannot rule out reverse causality and cannot be used for *in silico* screening. Regarding GEARS, neither different data splits nor different input graphs produced superior performance over a simple baseline, which was observed in the original study (Roohani et al., 2022). We remain uncertain about the cause of this discrepancy. To facilitate further comparisons, we have published all code, environment setup, and preprocessed datasets used in this work. Regarding GeneFormer, our results describe a possible extension and do not contradict any previously published results.

We conclude that generalizing to genes not perturbed in the training data is extremely challenging in practice. There is some theoretical basis for this: in certain frameworks, identifiability of causal effects requires perturbation of every gene (Hyttinen et al., 2012; Wagner,

2001). Other frameworks require fewer or no perturbations (Akutsu et al., 1999; Hashimoto, Gifford, & Jaakkola, 2016; Lopez, Hütter, et al., 2022); but, these guarantees require complete observation of direct causes of transcription, which is a strong assumption that has not held in the limited settings where it has been tested on transcriptome data (Kernfeld, Keener, Cahan, & Battle, 2023).

Despite these negative results, further methods development is warranted, because reliable *in silico* perturbation screens would be extremely valuable. Importantly, we distributed the perturbations such that a given gene was perturbed in only one of the test or training set. An alternative evaluation strategy would be to test on new combinations of genes or new doses of genes that were each perturbed individually or at a standard dose in the training set. Prediction of dose-dependent effects or genetic interactions is likely more feasible than predicting previously unperturbed genes, and we anticipate that DCD-FG, GEARS, and similar methods could be a valuable way to rationally extend single-factor or low-MOI screens. Sharing information across cell types is also promising, since thousands of genes have already been perturbed in certain cell lines (Replogle et al., 2022; Subramanian et al., 2017). In this vein, foundation models remain largely unexplored, and we specifically note that our feature extraction strategy is not the only way GeneFormer could have been used. Another area with promising individual demonstrations is expression forecasting from time-series or RNA velocity data (Burdziak et al., 2023; Kamimoto et al., 2023; Pemberton-Ross et al., 2015; Qiu et al., 2022; Yeo et al., 2021). However, in light of our results, claims of causal identification from transfer learning across cell types or from time-series data should be tested by thorough and neutral benchmarks that include maximally informative baseline methods.

Methods

Data and code availability

All project code is collected at https://github.com/ekernf01/perturbation_benchmarking. Our data collection is available on Zenodo at the DOI 10.5281/zenodo.8071809.

Simulated data

During simulations, we generated data from a vector autoregressive model using the following algorithm.

- Inputs:
    - D by D matrix M
    - N by D initial states X_0
    - Genes to perturb P
    - number of steps S
- Outputs:
    - Simulated expression profiles of size N by D for:
        - Each perturbation at the final time-point
        - Each control at the final timepoint
        - Each control's initial state

- Procedure:
    - For p in P:
        - $X_{0,p} = X_0$
        - For t in 1 … S:
            - $X_{t,p} = MX_{t-1,p}$
            - $X_{t,p}[:,P] = 1$
        - Return $X_{t,p}$
- Initial values
    - D=811, 713, 8, 788, or 663: the number of TF's in the CellOracle, gtex, MARA_FANTOM4, cellnet_human_Hg1332, or cellnet_human_Hugene networks, respectively
    - M_{ij} = c if i regulates j in the indicated network, else 0, with c scaled so the maximum eigenvalue of M is 0.01 (unless already 0)
    - $X_0 = 0$
    - P includes all genes
    - S=10

Perturbation data acquisition and preprocessing

The Nakatake data were acquired as normalized expression values by personal communication with ExAtlas maintainer Alexei Sharov. The Adamson and Dixit Perturb-seq data were obtained using GEARS. For all other datasets, source URL's and citations are provided in the metadata accompanying our data collection.

Our preprocessing typically includes pseudo-bulk aggregation, DESeq-2 normalization, and log1p transformation. For knockdown and overexpression experiments, we remove samples where directly perturbed genes do not change in the expected direction. For knockout experiments, no such filter is applied, since mRNA may still be produced and may even increase due to compensatory mechanisms.

Some dataset-specific procedures differ from the above steps. For the Nakatake data, missing values were replaced with control expression, and replication enabled testing of most perturbed genes, so we included samples only if the overexpressed gene saw a significant increase ($p<0.1$, t-test) or was not observed. For the Freimer data, we follow the original authors and remove the outlying sample Donor_4_AAVS1_6. Preprocessing for the Adamson and Dixit perturb-seq data follows GEARS (Roohani et al., 2022) to facilitate comparisons. To preprocess the Replogle overexpression data, we removed cells with under 2,000 UMIs or over 17,000 UMIs (99th percentile). We removed cells where over 40% of UMIs were from the 50 highest expressed genes, cells where over 20% of UMIs were from mitochondrially encoded genes, and cells where over 30% of UMIs were ribosomal protein subunits. We removed genes detected in fewer than 10 cells, unless they were overexpressed. Preprocessing for the Frangieh Perturb-CITE-seq includes four distinct versions. Version one follows Lopez et al. exactly to facilitate comparisons (Lopez, Hütter, et al., 2022). Version 2 additionally removes cells with total UMIs above the 99th percentile, with 6000 or fewer total UMIs, with 30% or more UMIs in the top 50 expressed genes, with 10% or more UMIs from mitochondrially encoded genes, with

20% or more UMIs from ribosomal protein subunits, or with two or more gRNAs detected. Version 3 follows version 2, but “pseudobulk”: we added together raw counts for all cells sharing a gRNA and applied total count normalization and log1p transformation. V4 is like V1, but cell cycle phase is regressed out. For the dataset labeled “replogle”, we remove cells with total UMIs above the 99th percentile, with 2000 or fewer total UMIs, with 40% or more UMIs in the top 50 expressed genes, with 20% or more UMIs from mitochondrially encoded genes, with 30% or more UMIs from ribosomal protein subunits.

Network acquisition and preprocessing

Source URL's, citations, and descriptions are given in the metadata accompanying our network collection. Most acquisition is fully automated using R or python. Fully dense networks from FNTM and Humanbase were subsetted to retain only edges with posterior probability over 50%, and fully dense networks from ANANSE_0.5 or ANANSE_0.8 were subsetted to retain edges with weight over 50% or 80% respectively. For CellNet, starting from the documentation at http://pcahan1.github.io/cellnetr/, we downloaded the network files:

cnProc_Hg1332_062414.R
cnProc_mogene_062414.R
cnProc_Hugene_062414.R
cnProc_mouse4302_062414.R

We installed cellnetr using the instructions at https://groups.google.com/forum/#!topic/cellnet_r/pXHt2J6ZH6I. The remainder of the processing is automated.

GRN inference methods

In initial evaluations, we use simple methods based on supervised machine learning. For each gene, a scikit-learn regressor is trained using the gene’s expression as labels and using the expression of certain TF’s as features. By default, all TF’s are included, using a manually curated list of TF’s (Lambert et al., 2018). But, if a network is used in the evaluation, then only network-adjacent TF’s are included as features. No gene is allowed to auto-regulate. Instances where gene A was perturbed directly are not used to train the model used to predict expression of gene A. To predict perturbation outcomes, we begin with the average expression of all negative controls. The corresponding transcription factor is then set to 0 (for knockout experiments) or to its observed value (for knockdown or overexpression experiments). From the resulting perturbed features, regression models are used to make predictions.

Within this same regression framework, GeneFormer is used to generate perturbed embeddings using in_silico_perturber.delete_index, in_silico_perturber.overexpress_index, and a forward pass through the model. This is used as a feature extraction step, yielding 256 features per observation. Raw counts with all genes, not normalized counts with variable genes, were passed to the GeneFormer tokenizer.

DCD-FG and GEARS are used according to the maintainers' documentation. NOTEARS and NOTEARS-LR are run using DCD-FG with the LinearGaussianModel and LinearModuleGaussianModel objects, and the constraint mode is always spectral_radius. GEARS is run using either the default GO graph or using a subgraph containing genes that are perturbed in the training set or the test set; these are specified using the gene_set_path argument in the PertData constructor.

Evaluations

We attempted to test every method on the same datasets presented in Fig. 3. Any omissions are due to runtime errors. In particular, GEARS requires replication of each perturbation condition and yields errors when only a single observation is available for some condition, and certain tree-based regression methods fail due to memory requirements exceeding 64GB.

For evaluation, our software offers two types of data split: "interventional" (used throughout unless otherwise noted) and "simple". For a "simple" split, a simple random sample was allocated to the test set. For an "interventional" split, samples were grouped by perturbation, and each perturbation was allocated to the training set or the test set. A 50-50 split means 50% of perturbations, but not necessarily 50% of observations, are allocated to the test set. For experiments involving prior network structures, perturbations are allocated to the test set only if they are present in at least one of the network structures.

GeneFormer was not pretrained on any of the datasets we use for evaluation.

For each test-set perturbation, we compare predicted and observed expression by evaluating spearman correlation, mean squared error, mean absolute error, mean absolute error on the (20, 100, 200) most differentially expressed genes, and the proportion of genes where the direction of change is correct. All measures were inspected but we typically present "MAE benefit", which is the mean absolute error subtracted from the mean absolute error attained by the mean of the training data.

Author contributions

E.K. initiated the study, harmonized the data, and conducted most of the benchmarking. Y.Y. benchmarked DCD-FG and performed due diligence on several perturbation datasets. J.W. tested code, interpreted results, and advised simulations. P.C. and A.B. funded and supervised the study. All authors edited the manuscript.

Acknowledgements

We are grateful to Romain Lopez, Yusuf Roohani, and Christina Theodoris for helpful advice on running DCD-FG, GEARS, and GeneFormer respectively. Thanks to Prashanthi Ravichandran and Emily Su for useful discussions. Thanks to Zexi Liu for helping test the code. AB was funded by NIH grant R35GM139580. PC was funded by NIH grant R35GM124725. Figure 2

includes clipart from the Noun Project: “crispr” by Cornelia Scheitz, “cell” by LAFS, and “rna” by Bacontaco. All are licensed under CC BY-SA 4.0.

Declaration of interests

A.B. is a stockholder for Alphabet, Inc, and has consulted for Third Rock Ventures.

Supplemental information

Supplemental file 1: Full documentation of the Grammar of Gene Regulatory Networks

Bibliography

Adamson, B., Norman, T. M., Jost, M., Cho, M. Y., Nuñez, J. K., Chen, Y., Villalta, J. E., et al. (2016). A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response. *Cell*, *167*(7), 1867-1882.e21.

Akutsu, T., Miyano, S., & Kuhara, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing*, 17–28.

Amit, I., Garber, M., Chevrier, N., Leite, A. P., Donner, Y., Eisenhaure, T., Guttman, M., et al. (2009). Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses. *Science*, *326*(5950), 257–263.

Amrute, J. M., Lai, L., Ma, P., Koenig, A. L., Kamimoto, K., Bredemeyer, A., Shankar, T. S., et al. (2022). Defining cardiac recovery at single cell resolution. *BioRxiv*, *Preprint posted online 9/13/2022*, 10.1101/2022.09.11.507463.

Baca, S. C., Takeda, D. Y., Seo, J.-H., Hwang, J., Ku, S. Y., Arafeh, R., Arnoff, T., et al. (2021). Reprogramming of the FOXA1 cistrome in treatment-emergent neuroendocrine prostate cancer. *Nature Communications*, *12*(1), 1979.

Badia-I-Mompel, P., Wessels, L., Müller-Dott, S., Trimbour, R., Ramirez Flores, R. O., Argelaguet, R., & Saez-Rodriguez, J. (2023). Gene regulatory network inference in the era

of single-cell multi-omics. *Nature Reviews. Genetics*.

Bonneau, R., Facciotti, M. T., Reiss, D. J., Schmid, A. K., Pan, M., Kaur, A., Thorsson, V., et al. (2007). A predictive model for transcriptional control of physiology in a free living cell. *Cell*, *131*(7), 1354–1365.

Bonneau, R., Reiss, D. J., Shannon, P., Facciotti, M., Hood, L., Baliga, N. S., & Thorsson, V. (2006). The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, *7*(5), R36.

Boulesteix, A.-L., Lauer, S., & Eugster, M. J. A. (2013). A plea for neutral comparison studies in computational sciences. *Plos One*, *8*(4), e61562.

Burdziak, C., Zhao, C. J., Haviv, D., Alonso-Curbelo, D., Lowe, S. W., & Pe'er, D. (2023). scKINETICS: inference of regulatory velocity with single-cell transcriptomics data. *Bioinformatics*, *39*(39 Suppl 1), i394–i403.

Chevalley, M., Mehrjou, A., Schwab, P., Notin, P., & Roohani, Y. (2022, March 1). Chevalley, M., Mehrjou, A., Schwab, P., Notin, P., & Roohani, Y. (n.d.). Causalbench Challenge. https://www.gsk.ai/causalbench-challenge/ . *Causalbench Challenge*. Retrieved June 6, 2023, from https://www.gsk.ai/causalbench-challenge/

Chevalley, M., Roohani, Y., Mehrjou, A., Leskovec, J., & Schwab, P. (2022). CausalBench: A Large-scale Benchmark for Network Inference from Single-cell Perturbation Data. *arXiv*.

Cui, H., Wang, C., Maan, H., & Wang, B. (2023). scGPT: Towards Building a Foundation Model for Single-Cell Multi-omics Using Generative AI. *BioRxiv*.

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C. P., Jerby-Arnon, L., Marjanovic, N. D., et al. (2016). Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell*, *167*(7), 1853-1866.e17.

Duggan, D. J., Bittner, M., Chen, Y., Meltzer, P., & Trent, J. M. (1999). Expression profiling using cDNA microarrays. *Nature Genetics*, *21*(1 Suppl), 10–14.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*,

63(1), 3–42.

Hashimoto, T., Gifford, D., & Jaakkola, T. (2016). Learning Population-Level Diffusions with Generative RNNs.

Hyttinen, A., Eberhardt, F., & Hoyer, P. O. (2012). Learning Linear Cyclic Causal Models with Latent Variables. *Journal of Machine Learning Research*.

Jiang, J., Chen, S., Tsou, T., McGinnis, C. S., Khazaei, T., Zhu, Q., Park, J. H., et al. (2023). D-SPIN constructs gene regulatory network models from multiplexed scRNA-seq data revealing organizing principles of cellular perturbation response. *BioRxiv*.

Kamal, A., Arnold, C., Claringbould, A., Moussa, R., Servaas, N. H., Kholmatov, M., Daga, N., et al. (2023). GRaNIE and GRaNPA: inference and evaluation of enhancer-mediated gene regulatory networks. *Molecular Systems Biology*, *19*(6), e11627.

Kamimoto, K., Hoffmann, C. M., & Morris, S. A. (2020). CellOracle: Dissecting cell identity via network inference and in silico gene perturbation. *BioRxiv*.

Kamimoto, K., Stringa, B., Hoffmann, C. M., Jindal, K., Solnica-Krezel, L., & Morris, S. A. (2023). Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, *614*(7949), 742–751.

Kernfeld, E. M., Keener, R. M., Cahan, P., & Battle, A. (2023). Model-X knockoffs reveal data-dependent limits on regulatory network identification. *BioRxiv*.

Lambert, S. A., Jolma, A., Campitelli, L. F., Das, P. K., Yin, Y., Albu, M., Chen, X., et al. (2018). The human transcription factors. *Cell*, *172*(4), 650–665.

Lasri, A., Shahrezaei, V., & Sturrock, M. (2022). Benchmarking imputation methods for network inference using a novel method of synthetic scRNA-seq data generation. *BMC Bioinformatics*, *23*(1), 236.

Lee, H.-Y., Jeon, Y., Kim, Y. K., Jang, J. Y., Cho, Y. S., Bhak, J., & Cho, K.-H. (2021). Identifying molecular targets for reverse aging using integrated network analysis of transcriptomic and epigenomic changes during aging. *Scientific Reports*, *11*(1), 12317.

Lek, M., Karczewski, K. J., Minikel, E. V., Samocha, K. E., Banks, E., Fennell, T., O'Donnell-Luria, A. H., et al. (2016). Analysis of protein-coding genetic variation in 60,706 humans. *Nature*, *536*(7616), 285–291.

Liang, S., Fuhrman, S., & Somogyi, R. (1998). Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing*, 18–29.

Lopez, R., Hütter, J.-C., Pritchard, J. K., & Regev, A. (2022). Large-Scale Differentiable Causal Discovery of Factor Graphs. *arXiv*.

Lopez, R., Tagasovska, N., Ra, S., Cho, K., Pritchard, J. K., & Regev, A. (2022). Learning Causal Representations of Single Cells via Sparse Mechanism Shift Modeling. *arXiv*.

Maathuis, M. H., Colombo, D., Kalisch, M., & Bühlmann, P. (2010). Predicting causal effects in large-scale systems from observational data. *Nature Methods*, *7*(4), 247–248.

Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., Allison, K. R., et al. (2012). Wisdom of crowds for robust gene network inference. *Nature Methods*, *9*(8), 796–804.

McCalla, S. G., Fotuhi Siahpirani, A., Li, J., Pyne, S., Stone, M., Periyasamy, V., Shin, J., et al. (2023). Identifying strengths and weaknesses of methods for computational network inference from single-cell RNA-seq data. *G3 (Bethesda, Md.)*, *13*(3).

Meinshausen, N., Hauser, A., Mooij, J. M., Peters, J., Versteeg, P., & Bühlmann, P. (2016). Methods for causal inference from gene perturbation experiments and validation. *Proceedings of the National Academy of Sciences of the United States of America*, *113*(27), 7361–7368.

Miraldi, E. R., Pokrovskii, M., Watters, A., Castro, D. M., De Veaux, N., Hall, J. A., Lee, J.-Y., et al. (2019). Leveraging chromatin accessibility for transcriptional regulatory network inference in T Helper 17 Cells. *Genome Research*, *29*(3), 449–463.

Nakatake, Y., Ko, S. B. H., Sharov, A. A., Wakabayashi, S., Murakami, M., Sakota, M., Chikazawa, N., et al. (2020). Generation and profiling of 2,135 human ESC lines for the

systematic analyses of cell states perturbed by inducing single transcription factors. *Cell reports*, *31*(7), 107655.

Nießl, C., Herrmann, M., Wiedemann, C., Casalicchio, G., & Boulesteix, A. (2022). Over‐optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results. *WIREs Data Mining and Knowledge Discovery*, *12*(2).

Norman, T. M., Horlbeck, M. A., Replogle, J. M., Ge, A. Y., Xu, A., Jost, M., Gilbert, L. A., et al. (2019). Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, *365*(6455), 786–793.

Osorio, D., Zhong, Y., Li, G., Xu, Q., Yang, Y., Tian, Y., Chapkin, R. S., et al. (2022). scTenifoldKnk: An efficient virtual knockout tool for gene function predictions via single-cell gene regulatory network perturbation. *Patterns (New York, N.Y.)*, *3*(3), 100434.

Pawel, S., Kook, L., & Reeve, K. (2023). Pitfalls and potentials in simulation studies: Questionable research practices in comparative simulation studies allow for spurious claims of superiority of any method. *Biometrical Journal. Biometrische Zeitschrift*, e2200091.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*.

Pemberton-Ross, P. J., Pachkov, M., & van Nimwegen, E. (2015). ARMADA: Using motif activity dynamics to infer gene regulatory networks from gene expression data. *Methods*, *85*, 62–74.

Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A., & Murali, T. M. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature Methods*, *17*(2), 147–154.

Qiu, X., Zhang, Y., Martin-Rufino, J. D., Weng, C., Hosseinzadeh, S., Yang, D., Pogson, A. N., et al. (2022). Mapping transcriptomic vector fields of single cells. *Cell*, *185*(4), 690-711.e45.

Reddy, J., Fonseca, M. A. S., Corona, R. I., Nameki, R., Segato Dezem, F., Klein, I. A., Chang, H., et al. (2021). Predicting master transcription factors from pan-cancer expression data.

Science Advances, *7*(48), eabf6123.

Replogle, J. M., Norman, T. M., Xu, A., Hussmann, J. A., Chen, J., Cogan, J. Z., Meer, E. J., et al. (2020). Combinatorial single-cell CRISPR screens by direct guide RNA capture and targeted sequencing. *Nature Biotechnology*, *38*(8), 954–961.

Replogle, J. M., Saunders, R. A., Pogson, A. N., Hussmann, J. A., Lenail, A., Guna, A., Mascibroda, L., et al. (2022). Mapping information-rich genotype-phenotype landscapes with genome-scale Perturb-seq. *Cell*, *185*(14), 2559-2575.e28.

Roohani, Y., Huang, K., & Leskovec, J. (2022). GEARS: Predicting transcriptional outcomes of novel multi-gene perturbations. *BioRxiv*.

Saint-Antoine, M., & Singh, A. (2023). Benchmarking gene regulatory network inference methods on simulated and experimental data. *BioRxiv*.

Subramanian, A., Narayan, R., Corsello, S. M., Peck, D. D., Natoli, T. E., Lu, X., Gould, J., et al. (2017). A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell*, *171*(6), 1437-1452.e17.

Theodoris, C. V., Xiao, L., Chopra, A., Chaffin, M. D., Al Sayed, Z. R., Hill, M. C., Mantineo, H., et al. (2023). Transfer learning enables predictions in network biology. *Nature*, *618*(7965), 616–624.

Tran, A., Yang, P., Yang, J. Y. H., & Ormerod, J. T. (2022). scREMOTE: Using multimodal single cell data to predict regulatory gene relationships and to build a computational cell reprogramming model. *NAR Genomics and Bioinformatics*, *4*(1), lqac023.

Wagner, A. (2001). How to reconstruct a large genetic network from n gene perturbations in fewer than n(2) easy steps. *Bioinformatics*, *17*(12), 1183–1197.

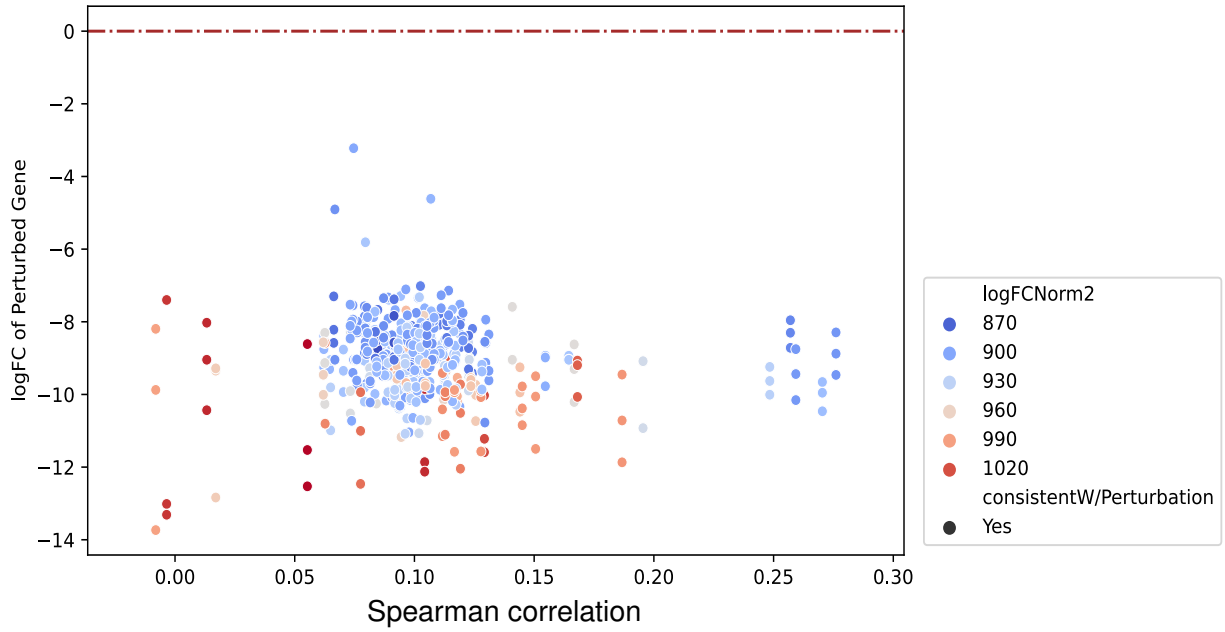
Wang, L., Trasanidis, N., Wu, T., Dong, G., Hu, M., Bauer, D. E., & Pinello, L. (2022). Dictys: dynamic gene regulatory network dissects developmental continuum with single-cell multi-omics. *BioRxiv*.

Yeo, G. H. T., Saksena, S. D., & Gifford, D. K. (2021). Generative modeling of single-cell time

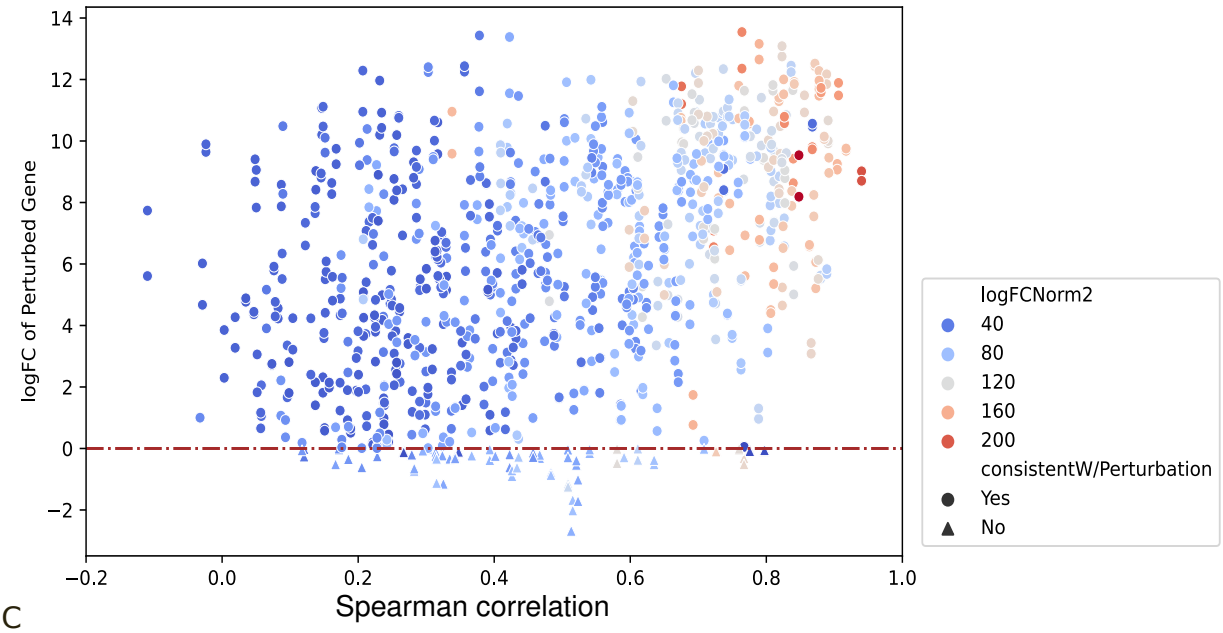
series with PRESCIENT enables prediction of cell trajectories with interventions. *Nature Communications*, *12*(1), 3222.

Young, W. C., Raftery, A. E., & Yeung, K. Y. (2014). Fast Bayesian inference for gene regulatory networks using ScanBMA. *BMC Systems Biology*, *8*, 47.

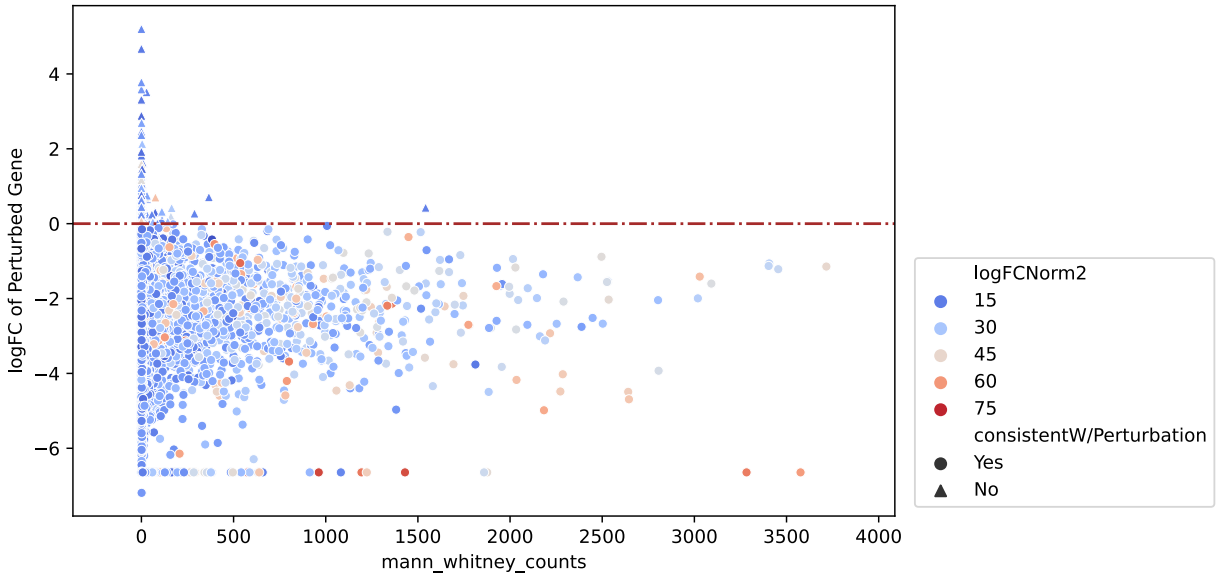
A


logFC of Perturbed Gene
Spearman correlation
logFCNorm2
870
900
930
960
990
1020
consistentW/Perturbation
Yes

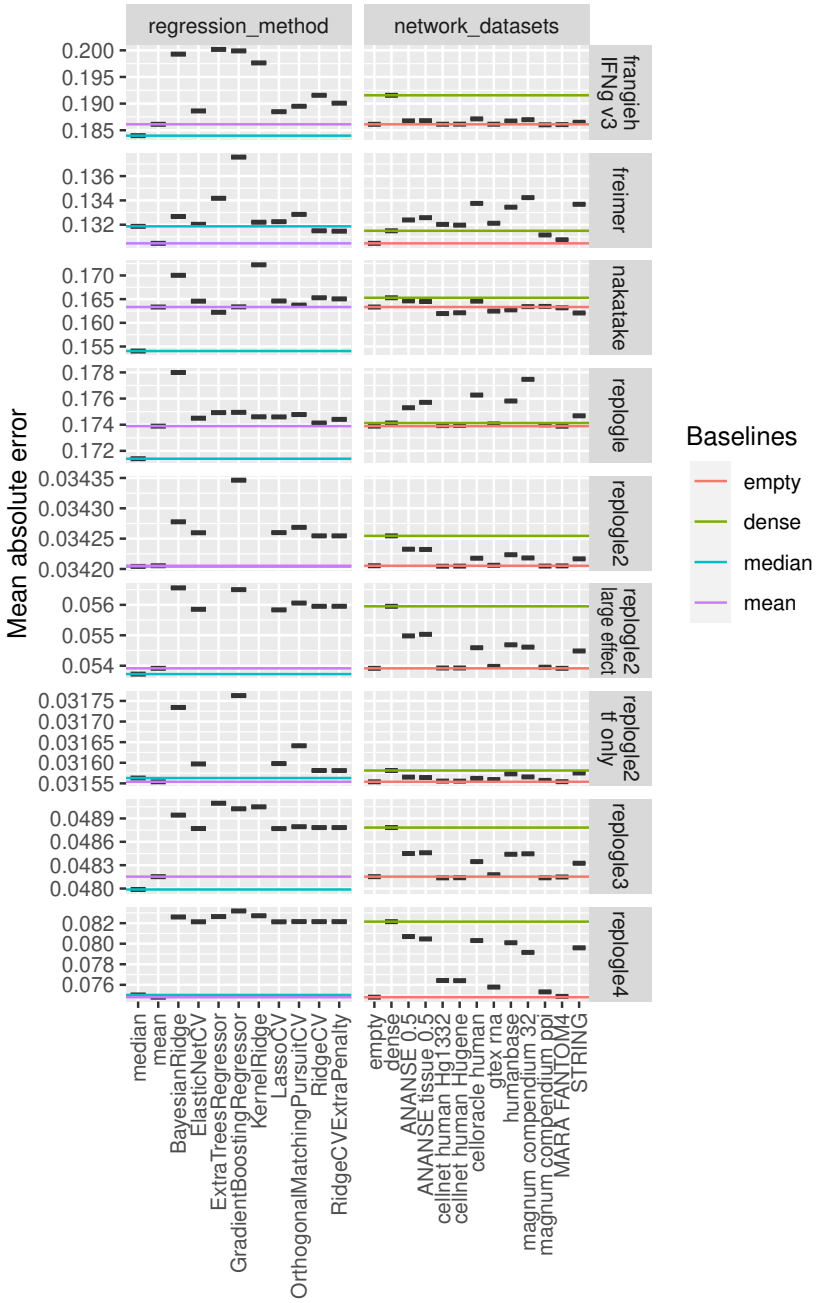
B


logFC of Perturbed Gene
Spearman correlation
logFCNorm2
40
80
120
160
200
consistentW/Perturbation
Yes
No

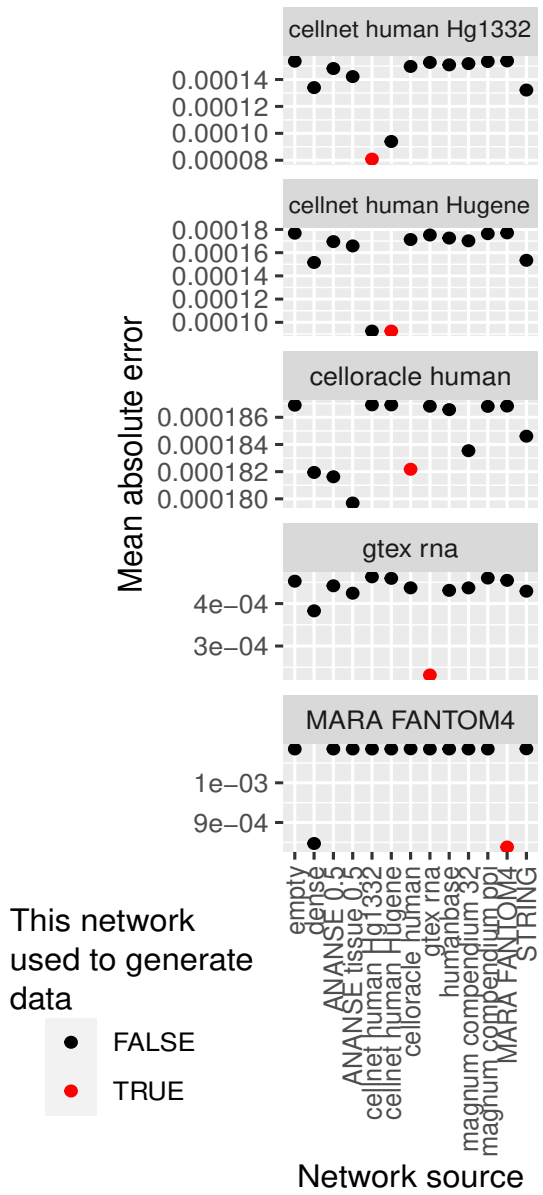
C


logFC of Perturbed Gene
mann_whitney_counts
logFCNorm2
15
30
45
60
75
consistentW/Perturbation
Yes
No

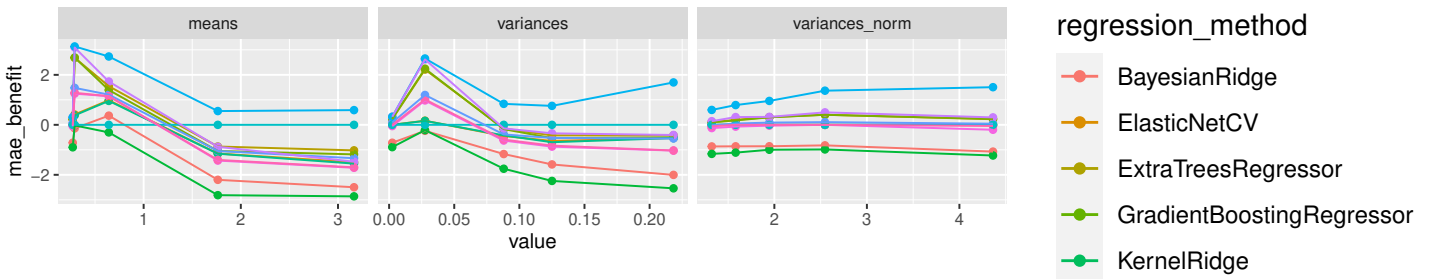
A


regression_method
network_datasets
Mean absolute error
frangieh IFNg v3
freimer
nakatake
replogle
replogle2
replogle2 large effect
replogle2 tf only
replogle3
replogle4
median
mean
BayesianRidge
ElasticNetCV
ExtraTreesRegressor
GradientBoostingRegressor
KernelRidge
LassoCV
OrthogonalMatchingPursuitCV
RidgeCV
RidgeCVExtraPenalty
empty
dense
ANANSE 0.5
ANANSE tissue 0.5
cellnet human Hg1332
cellnet human Hugene
celloracle human
gtex rna
humanbase
magnum compendium 32
magnum compendium ppi
MARA FANTOM4
STRING
Baselines
empty
dense
median
mean

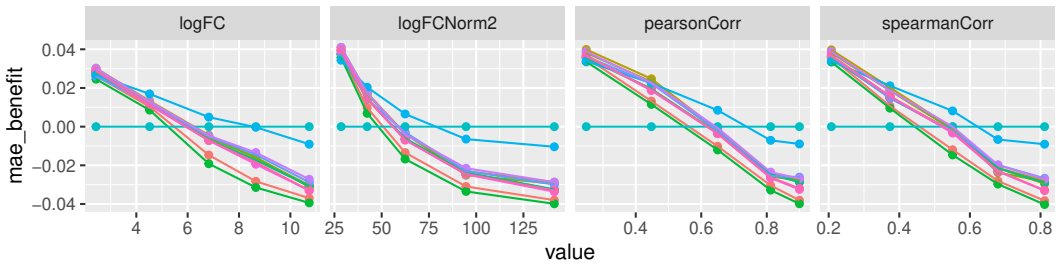
B

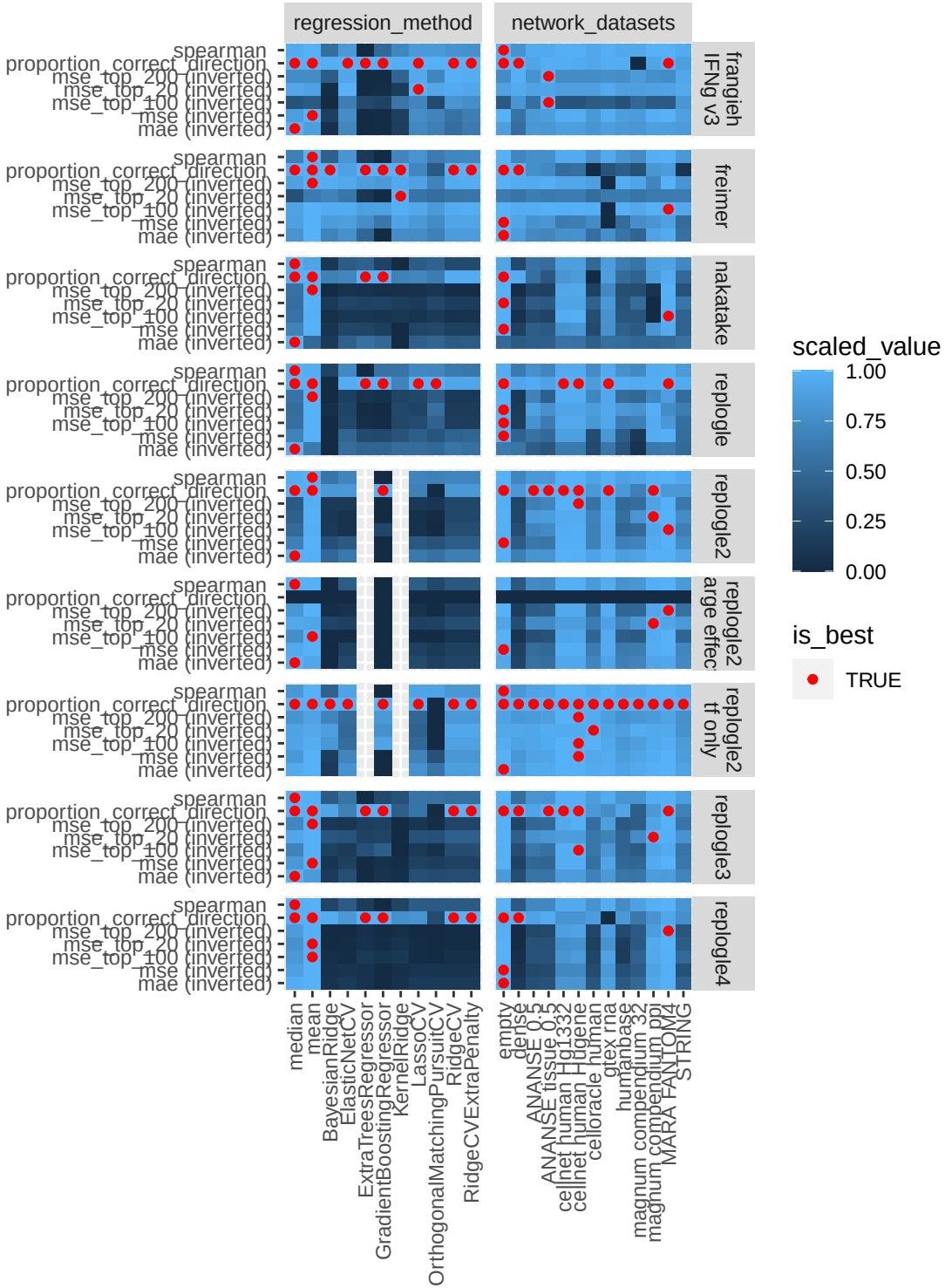

cellnet human Hg1332
cellnet human Hugene
celloracle human
gtex rna
MARA FANTOM4
Mean absolute error
empty
dense
ANANSE 0.5
ANANSE tissue 0.5
cellnet human Hg1332
cellnet human Hugene
celloracle human
gtex rna
humanbase
magnum compendium 32
magnum compendium ppi
MARA FANTOM4
STRING
Network source
This network used to generate data
FALSE
TRUE

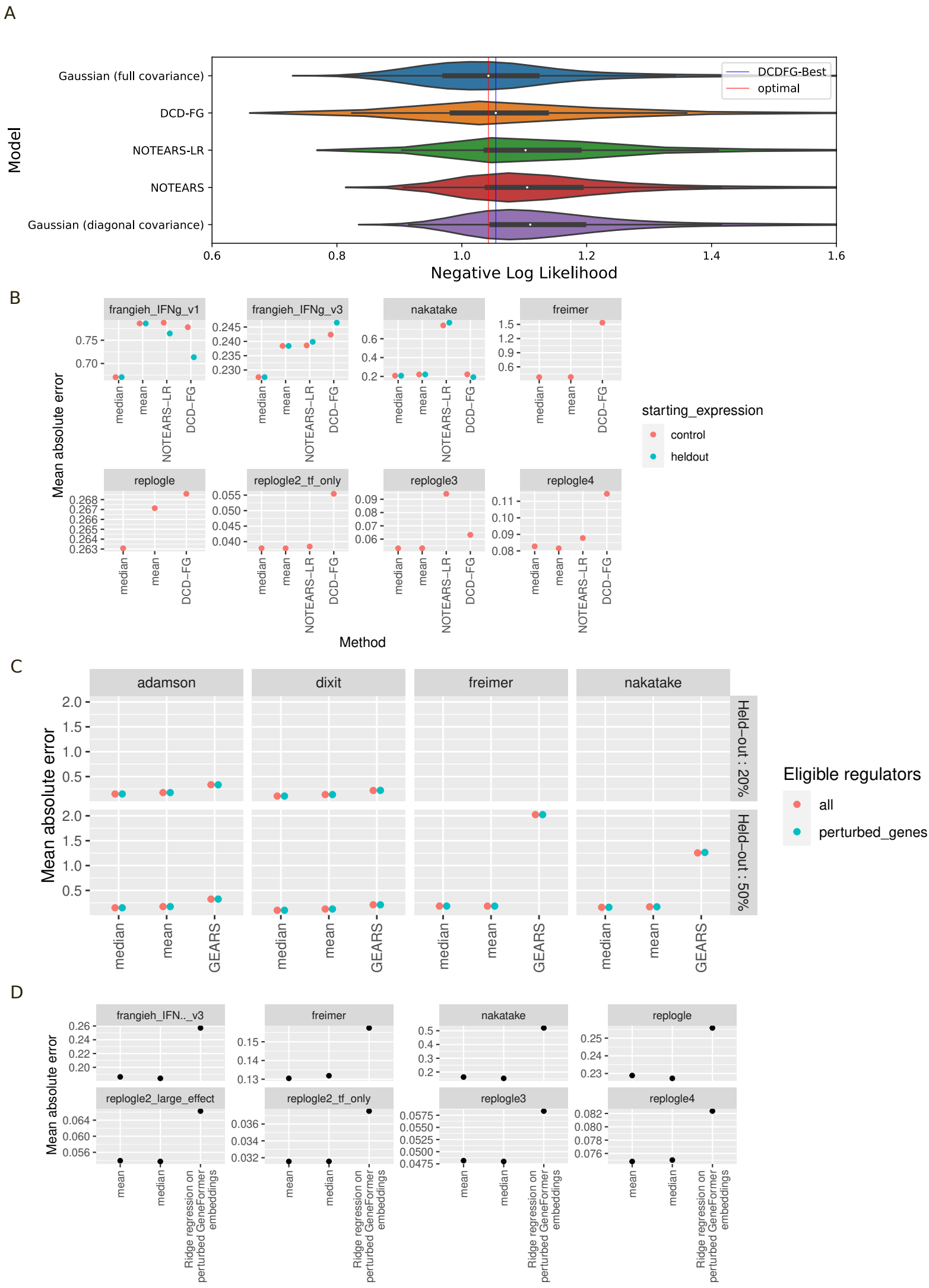
C


means
variances
variances_norm
mae_benefit
value
regression_method
BayesianRidge
ElasticNetCV
ExtraTreesRegressor
GradientBoostingRegressor
KernelRidge
LassoCV
mean
median
OrthogonalMatchingPursuitCV
RidgeCV
RidgeCVExtraPenalty

D


logFC
logFCNorm2
pearsonCorr
spearmanCorr
mae_benefit
value


regression_method
network_datasets
frangieh IFNg v3
freimer
nakatake
replogle
replogle2
replogle2 arge effec
replogle2 tf only
replogle3
replogle4
spearman
proportion_correct_direction
mse_top_200 (inverted)
mse_top_20 (inverted)
mse_top_100 (inverted)
mse (inverted)
mae (inverted)
median
mean
BayesianRidge
ElasticNetCV
ExtraTreesRegressor
GradientBoostingRegressor
KernelRidge
LassoCV
OrthogonalMatchingPursuitCV
RidgeCV
RidgeCVExtraPenalty
empty
dense
ANANSE 0.5
ANANSE tissue 0.5
cellnet human Hg1332
cellnet human Hugene
celloracle_human
gtex_rna
humanbase
magnum_compendium_32
magnum_compendium_poi
MARA FANTOM4
STRING
scaled_value
1.00
0.75
0.50
0.25
0.00
is_best
TRUE


A
Model
Gaussian (full covariance)
DCD-FG
NOTEARS-LR
NOTEARS
Gaussian (diagonal covariance)
DCDFG-Best
optimal
0.6
0.8
1.0
1.2
1.4
1.6
Negative Log Likelihood
B
frangieh_IFNg_v1
frangieh_IFNg_v3
nakatake
freimer
replogle
replogle2_tf_only
replogle3
replogle4
Mean absolute error
median
mean
NOTEARS–LR
DCD–FG
Method
starting_expression
control
heldout
C
adamson
dixit
freimer
nakatake
Held–out : 20%
Held–out : 50%
Mean absolute error
median
mean
GEARS
Eligible regulators
all
perturbed_genes
D
frangieh_IFN.._v3
freimer
nakatake
replogle
replogle2_large_effect
replogle2_tf_only
replogle3
replogle4
Mean absolute error
mean
median
Ridge regression on perturbed GeneFormer embeddings