Multi-study inference of regulatory networks for more accurate models of gene regulation

Dayanne M. Castro[1], Nicholas R. De Veaux[2], Emily R. Miraldi[3,4], Richard Bonneau[1,2]

1 New York University, New York, NY 10003, USA

2 Center for Computational Biology, Flatiron Institute, New York, NY 10010, USA

3 Department of Pediatrics, University of Cincinnati College of Medicine, Cincinnati, OH 45229, USA

4 Divisions of Immunobiology & Biomedical Informatics, Cincinnati Children's Hospital, Cincinnati, OH 45229, USA

* rb133@nyu.edu

Abstract

Gene regulatory networks are composed of sub-networks that are often shared across biological processes, cell-types, and organisms. Leveraging multiple sources of information, such as publicly available gene expression datasets, could therefore be helpful when learning a network of interest. Integrating data across different studies, however, raises numerous technical concerns. Hence, a common approach in network inference, and broadly in genomics research, is to separately learn models from each dataset and combine the results. Individual models, however, often suffer from under-sampling, poor generalization and limited network recovery. In this study, we explore previous integration strategies, such as batch-correction and model ensembles, and introduce a new multitask learning approach for joint network inference across several datasets. Our method initially estimates the activities of transcription factors, and subsequently, infers the relevant network topology. As regulatory interactions are context-dependent, we estimate model coefficients as a combination of both dataset-specific and conserved components. In addition, adaptive penalties may be used to favor models that include interactions derived from multiple sources of prior knowledge including orthogonal genomics experiments. We evaluate generalization and network recovery using examples from *Bacillus subtilis* and *Saccharomyces cerevisiae*, and show that sharing information across models improves network reconstruction. Finally, we demonstrate robustness to both false positives in the prior information and heterogeneity among datasets.

Introduction

Gene regulatory network inference aims at computationally deriving and ranking regulatory hypotheses on transcription factor-target gene interactions [1–3]. Often, these regulatory models are learned from gene expression measurements across a large number of samples. Strategies to obtain such data range from combining several publicly available datasets to generating large expression datasets from scratch [4–7]. Given decreasing costs of sequencing and the exponential growth in the availability of gene expression data in public databases [8, 9], data integration across several studies becomes particularly promising for an increasing number of biological systems.

In theory, multi-study analyses provide a better representation of the underlying cellular regulatory network, possibly revealing insights that could not be uncovered from individual studies [6]. In practice, however, biological datasets are highly susceptible to batch effects [10], which are systematic sources of technical variation due to different reagents, machines, handlers etc. that complicate omics meta-analyses [11, 12]. Although several methods to remove batch effects from expression data have been developed, they often rely on evenly distributed experimental designs across batches [13, 14]. Batch-correction methods may deflate relevant biological variability or induce incorrect differences between experimental groups when conditions are unbalanced across batches, which can significantly affect downstream analyses [15]. Therefore these batch effect removal methods are not applicable when integrating public data from multiple sources with widely differing experimental designs.

In network inference, an approach often taken to bypass batch effects is to learn models from each dataset separately and combine the resulting networks [16, 17]. Known as ensemble learning, this idea of synthesizing several weaker models into a stronger aggregate model is commonly used in machine learning to prevent overfitting and build more generalizable prediction models [18]. In several scenarios, ensemble learning avoids introducing additional artifacts and complexity that may be introduced by explicitly modeling batch effects. On the other hand, the relative sample size of each dataset is smaller when using ensemble methods, likely decreasing the ability of an algorithm to detect relevant interactions. As regulatory networks are highly context-dependent [19], for example, TF binding to several promoters is condition-specific [20], a drawback for both batch-correction and ensemble methods is that they produce a single network model to explain the data across datasets. Relevant dataset-specific interactions might not be recovered, or just difficult to tell apart using a single model.

Although it will not be the primary focus of this paper, most modern network inference algorithms integrate multiple data-types to derive prior or constraints on network structure. These priors/constraints have been shown to dramatically improve network model selection performance when combined with the state variables provided by expression data. In these methods [17, 21], priors or constraints on network structure (derived from multiple sources like known interactions, ATAC-seq, DHS, or ChIP-seq experiments [22–24]) are used to influence the penalty on adding model components, where edges in the prior are effectively penalized less. Here we describe a method that builds on that work (and similar work in other fields), but in addition we let model inference processes (each carried out using a separate data-set) influence each others model penalties, so that edges that agree across inference tasks are more likely to be uncovered [25–31]. Several previous works on this front focused on enforcing similarity across models by penalizing differences on strength and direction of regulatory interactions using a fusion penalty [25, 27, 28]. Because the influence of regulators on the expression of targets may vary across datasets, possibly even due to differences in measurement technologies, we look to induce similarity on network structure (the choice of regulators) using a group-sparse penalty. Previous methods also applied this type of penalty [26, 29, 31], however, they were not robust to differences in relevant edges across datasets.

Here we propose a multitask learning (MTL) approach to exploit cross-dataset commonalities while recognizing differences and is able to incorporate prior knowledge on network structure if available [32, 33]. In this framework, information flow across datasets leads the algorithm to prefer solutions that better generalize across domains, thus reducing chances of overfitting and improving model predictive power [34]. Since biological datasets are often under-sampled, we hypothesize that sharing information across models inferred from multiple datasets using a explicit multitask learning framework will improve accuracy of inferred network models in a variety of common experimental designs/settings.

In this paper, we explicitly show that joint inference significantly improves network recovery using examples from two model organisms, *Bacillus subtilis* and *Saccharomyces cerevisiae*. We show that models inferred for each dataset using our MTL approach (which adaptively penalizes conserved and data-set-unique model components separately) are vastly more accurate than models inferred separately using a single-task learning (STL) approach. We also explore commonly used data integration strategies, and show that MTL outperforms both batch-correction and ensemble approaches. In addition, we also demonstrate that our method is robust to noise in the input prior information. Finally, we look at conserved and dataset-specific inferred interactions, and show that our method can leverage cross-dataset commonalities, while being robust to differences.

Results

Overview of network inference algorithm

To improve regulatory network inference from expression data, we developed a framework that leverages training signals across related expression datasets. For each gene, we assume that its regulators may overlap across conditions in related datasets, and thus we could increase our ability to uncover accurate regulatory interactions by inferring them jointly. Our method takes as input multiple expression datasets and priors on network structure, and then outputs regulatory hypotheses associated with a confidence score proportional to our belief that each prediction is true (Fig 1A). As previous studies [17, 35–37], our method also includes an intermediate step that estimates transcription factor activities (TFA), and then, models gene expression as a function of those estimates (Fig 1B).

In our model, TFA represent a relative quantification of active protein that is inducing or repressing the transcription of its targets in a given sample, and is an attempt to abstract away unmeasured factors that influence TFA in a living cell [37–39], such as post-translational regulation [40], protein-protein interactions [41], and chromatin accessibility [42]. We estimate TFA from partial knowledge of the network topology (Fig 1C) [21, 43–47] and gene expression data as previously proposed (Fig 1D) [17]. This is comparable to using a TF's targets collectively as a reporter for its activity.

Next, we learn the dependencies between gene expression and TFA and score predicted interactions. In this step, our method departs from previous work, and we employ a multitask learning to learn regulatory models across datasets jointly, as opposed to single-task learning, where network inference is performed for each dataset independently (Fig 1E). As genes are known to be regulated by a small number of TFs [48], we can assume that these models are sparse, that is, they contain only a few nonzero entries [3]. We thus implement both approaches using sparsity-inducing penalties derived from the lasso [49]. Here the network model is represented as a matrix for each target gene (where columns are data-sets/cell-types/studies and rows are potential regulators) with signed entries corresponding to strength and type of regulation.

Importantly, our MTL approach decomposes this model coefficients matrix into a dataset-specific component and a conserved component to enable us to penalize dataset-unique and conserved interactions separately for each target gene [32]; this separation captures differences in regulatory networks across datasets (Fig 2). Specifically, we apply an l_1/l_∞ penalty to the one component to encourage similarity between network models [50], and an l_1/l_1 penalty to the other to accommodate differences [32]. We also incorporate prior knowledge by using adaptive weights when penalizing different coefficients in the l_1/l_1 penalty [33]. Finally, we perform this step for several bootstraps of the conditions in the expression and activities matrices, and calculate a confidence score for each predicted interaction that represents both the stability across bootstraps and the proportion of variance explained of the target expression dependent on each predictor.

Our method is readily available in an open-source package, *Inferelator-AMuSR* (**A**daptive **Mu**ltiple **S**parse **R**egression), enabling TF activity estimation and multi-source gene regulatory network inference, ultimately facilitating mechanistic interpretations of gene expression data to the Biology community. In addition, this method allows for adaptive penalties to favor interactions with prior knowledge proportional to the user-defined belief that interactions in the prior are true. Finally, our implementation also includes several mechanisms that speed-up computations, making it scalable for the datasets here used, and support for parallel computing across multiple nodes and cores in several computing environments.

Model organisms, expression datasets, and priors

We validated our approach using two model organisms, a gram-positive bacteria, *B. subtilis*, and an eukaryote, *S. cerevisiae*. Availability of validated TF-target regulatory interactions, hereafter referred to as the gold-standard, make both organisms a good choice for exploring inference methods (3040 interactions, connecting 153 TFs to 1822 target genes for *B. subtilis* [17, 46], 1198 interactions connecting 91 TFs to 842 targets for *S. cerevisiae* [51]). For *B. subtilis*, we use two expression datasets. The first one, *B. subtilis 1*, was collected for strain PY79 and contains multiple knockouts, competence and sporulation-inducing conditions, and chemical treatments (429 samples, 38 experimental designs with multiple time-series experiments) [17]. The second dataset, *B. subtilis 2*, was collected for strain BSB1 and contains several nutritional, and other environmental stresses, as well as competence and sporulation-inducing conditions (269 samples, and 104 conditions) [52]. For *S. cerevisiae*, we downloaded three expression datasets from the SPELL database [53]. *S. cerevisiae 1* is a compendium of steady-state chemostat cultures with several combinations of cultivation parameters (170 samples, 55 conditions) [54]. *S. cerevisiae 2* profiles two yeast strains (BY and RM) grown with two carbon sources, glucose and ethanol, in different concentrations (246 samples, and 109 conditions) [55]. Finally, *S. cerevisiae 3* with expression profiles following several mutations and chemical treatments (300 samples) [56]. Each dataset was collected using a different microarray platform. Cross-platform data aggregation is well known to cause of strong batch effects [10]. For each species, we considered the set of genes present across datasets.

In our inference framework, prior knowledge on network topology is essential to first estimate transcription factor activities and to then bias model selection towards interactions with prior information during the network inference stage of the algorithm. Therefore, to properly evaluate our method, it is necessary to gather prior interactions independent of the ones in the gold-standard. For *B. subtilis*, we adopt the previously used strategy of partitioning the initial gold-standard into two disjoint sets, a prior for use in network inference and a gold-standard to evaluate model quality [17]. For *S. cerevisiae*, on the other hand, we wanted to explore a more realistic scenario, where a gold-standard is often not available. In the absence of such information, we hypothesized that orthogonal high-throughput datasets would provide insight. Because the yeast gold-standard [51] was built as a combination of TF-binding (ChIP-seq, ChIP-ChIP) and TF knockout datasets available in the YEASTRACT [47] and the SGD [57] databases, we propose to derive prior knowledge from chromatin accessibility data [22, 23] and TF binding sites [58] (as this is a realistic and efficient genomic experimental design for non-model organisms). Open regions in the genome can be scanned for transcription factor binding sites, which can provide an indirect evidence of regulatory function [59]. We then assigned TFs to the closest upstream gene, and built a prior matrix where entries represent the number of motifs for a particular TF that was associated to a gene [60]. We obtained a list of regulators from the YeastMine database [61], which we also used to sign entries in the prior: interactions for regulators described as repressors were marked as negative. Because genome-wide measurements of DNA accessibility can be obtained in a single experiment, using techniques that take advantage of the sensitivity of nucleosome-free DNA to endonuclease digestion (DNase-seq) or to Tn5 transposase insertion (ATAC-seq) [62], we expect this approach to be generalizable to several biological systems.

Sharing information across network models via multitask learning improves model accuracy

Using the above expression datasets and priors, we learn regulatory networks for each organism employing both single-task and our multitask approaches. To provide an intuition for cross-dataset transfer of knowledge, we compare confidence scores attributed to a single gold-standard interaction using either STL or MTL for each organism. For *B. subtilis*, we look at the interaction between the TF *sigD* and the gene *lytA* (Fig 3A). The relationship between the *sigD* activity and *lytA* expression in the first dataset *B. subtilis 1* is weaker than in *B. subtilis 2*. This is reflected in the predicted confidence scores, half as strong for *B. subtilis 1* than for *B. subtilis 2*, when each dataset is used separately to learn networks through STL. On the other hand, when we learn these networks in the MTL framework, information flows from *B. subtilis 2* to *B. subtilis 1*, and we assign a high confidence score to this interaction in both networks. Similarly, for *S. cerevisiae*, we look at the interaction between the TF *Msn2* and the target gene *Hsp104* (Fig 3B). In this particular case, we observe a stronger and easier-to-uncover relationship between *Msn2* estimated activity and *Hsp104* expression as the size of the dataset increases. Using STL, we assign a nonzero confidence score to this interaction for all datasets, although these are much smaller than the scores attributed when networks are learned using MTL.

Following these examples, we examined changes in confidence scores attributed to all interactions in the gold-standard in STL- and MTL-inferred networks (Fig 3C). Notably, we see a high level of synergy between the *B. subtilis* datasets. Lots of interactions missed by STL receive nonzero confidence scores through the MTL approach. For yeast, we observe major gains of gold-standard interactions in particular for *S. cerevisiae 1*, which is the dataset with the lowest number of samples. For datasets with larger sample size, *S. cerevisiae 2* and *S. cerevisiae 3*, we do not see similar synergy between datasets as in the *B. subtilis* datasets, suggesting higher heterogeneity across the yeast datasets.

In order to evaluate the overall quality of the inferred networks, we use area under precision-recall curves (AUPR) [16], widely used to quantify a classifier's ability to distinguish two classes and to rank predictions. Networks learned using MTL are significantly more accurate than networks learned using the STL approach. For *B. subtilis* (Fig 3D), we observe a 2-fold gain in AUPR, indicating significant complementarity between the datasets. For *S. cerevisiae* (Fig 3E), we observe a clear increase in performance for networks inferred for every dataset, indicating that our method is very robust to both data heterogeneity and potential false edges derived from chromatin accessibility in the prior.

Benefits of multitask learning exceed those from batch-correction and ensemble methods

Next, we asked whether the higher performance of the MTL framework could be achieved by other commonly used data integration strategies, such as batch-correction and ensemble methods. Ensemble methods include several algebraic combinations of predictions from separate classifiers trained within a single-domain (sum, mean, maximum, minimum [63]).

To address this question, we evaluated networks inferred using all available data. First, we combined regulatory models inferred for each dataset either through STL or MTL by taking the maximum inferred confidence score for each interaction, generating two networks hereafter called STL-C and MTL-C. Although taking the average is more commonly done (in particular, when multiple algorithms are applied to the same dataset, which is not the case here) [16], this would emphasize particularly the commonalities across datasets. In addition, the motivation to use an MTL framework is to increase statistical power, while maintaining separate models for each dataset, hopefully improving interpretability. For each organism, we also merged all datasets into one, and applied ComBat for batch-correction [64], because of its perceived higher performance [65]. We then learn network models from these larger batch-corrected datasets, STL-BC. Both for *B. subtilis* (Fig 4A) and *S. cerevisiae* (Fig 4B), the MTL-C networks significantly outperform the STL-C and STL-BC networks, indicating that cross-dataset information sharing during modelling is a better approach to integrate datasets from different domains. Interestingly, the STL-BC networks' increase in performance, as compared to STL networks in Fig 3, was more pronounced in yeast than in *B. subtilis.* We speculate that the higher overlap between the conditions in the two *B. subtilis* datasets led to lower additional information when merging them together. Moreover, batch-correcting in this scenario may have decreased dataset-specific variability. For yeast, on the other hand, conditions were very different across datasets, and much new information is gained by merging them into one. However, because of this very fact, it is likely that incorrect relationships between genes were induced as an artifact, possibly confounding the inference.

Our method is robust to increasing prior weights and noise in prior

Because genes are frequently co-regulated, and biological networks are redundant and robust to perturbations, spurious correlations between transcription factors and genes are highly prevalent in gene expression data [66, 67]. To help discriminate true from false interactions, it is essential to incorporate prior information to bias model selection towards interactions with prior knowledge. Indeed, incorporating prior knowledge has been shown shown to increase accuracy of inferred models in several studies [3, 21, 68].

For example, suppose that two regulators present highly correlated activities, but regulate different sets of genes. A regression-based model would be unable to differentiate between them, and only other sources of information, such as binding evidence nearby a target gene, could help selecting one predictor over the other in a principled way. Thus, we provide an option to integrate prior knowledge to our MTL approach in the model selection step by allowing the user to input a "prior weight". This weight is used to increase presence of prior interactions to the final model, and should be proportional to the quality of the input prior.

Sources of prior information for the two model organisms used in this study are fundamentally different. The *B. subtilis* prior is high-quality, derived from small-scale experiments, whereas the *S. cerevisiae* prior is noisier, likely with both high false-positive and false-negative rates, derived from high-throughput chromatin accessibility experiments and TF binding motifs. To understand differences in prior influences for the same organism, we also include the yeast gold-standard as a possible source of prior in this analysis. The number of TFs per target gene in the *B. subtilis* (Fig 5A) and the *S. cerevisiae* (Fig 5B) gold-standards (GS) is hardly ever greater than 2, with median of 1, whereas for the chromatin accessibility-derived priors (ATAC) for *S. cerevisiae*, the median is 11 (Fig 5C). A large number of regulators per gene likely indicates a high false-positive rate in the yeast ATAC prior. Given the differences in prior quality, we test the sensitivity of our method to the prior weight parameter. We applied increasing prior weight, and measured how the confidence scores attributed to prior interactions was affected (Fig 5C) for the three described above source of priors. Interestingly, the confidence scores distributions show dependencies on both the prior quality and the prior weights. When the gold-standard interactions for *B. subtilis* and *S. cerevisiae* are used as prior knowledge, they receive significantly higher scores than interactions in the *S. cerevisiae* chromatin accessibility-derived prior, which is proportional to our belief on the quality of the input prior information. Importantly, even when we set the prior weight value to a very high value, such as 10, interactions in the ATAC prior are not pushed to very high confidence scores, suggesting that our method is robust to the presence of false interactions in the prior.

Joint network inference is robust to dataset heterogeneity

Because multitask learning approaches are inclined to return models that are more similar to each other, we sought to understand how heterogeneity among datasets affected the inferred networks. Specifically, we quantified the overlap between the networks learned for each dataset for *B. subtilis* and yeast. That is, the number of edges that are unique or shared across networks inferred for each dataset (Fig 6). In this analysis, we consider valid only predictions within a 0.5 precision cut-off, calculated using only TFs and genes present in the gold-standard. Since the *B. subtilis* datasets share more conditions than the yeast datasets, we hypothesized that the *B. subtilis* networks would have a higher overlap than the yeast networks. As expected, we observe that about 50% of the total edges are shared among two *B. subtilis* networks (Fig 6A), whereas for yeast only about 31% (Fig 6B) and 35% (Fig 6C), using gold-standard and chromatin accessibility-derived priors respectively, of the total number of edges is shared by at least two of the three inferred networks. Therefore, our approach for joint inference is robust to cross-dataset influences, preserving relative uniqueness when datasets are more heterogeneous.

Discussion

In this study, we presented a multitask learning approach for joint inference of gene regulatory networks across multiple expression datasets that improves performance and biological interpretation by factoring network models derived from multiple datasets into conserved and dataset-specific components. Our approach is designed to leverage cross-dataset commonalities while preserving relevant differences. While other multitask methods for network inference penalize for differences in model coefficients across datasets [25–28, 30], our method leverages shared underlying topology rather than the influence of TFs on targets. We expect this method to be more robust, because, in living cells, a TF's influence on a gene's expression can change in different conditions. In addition, previous methods either deal with dataset-specific interactions [25], or apply proper sparsity inducing regularization penalties [26–30]. Our approach, on the other hand, addresses both concerns. Finally, we implemented an additional feature to allow for incorporation of prior knowledge on network topology in the model selection step.

Using two different model organisms, *B. subtilis* and *S. cerevisiae*, we show that joint inference results in accurate network models. We also show that multitask learning leads to more accurate models than other data integration strategies, such as batch-correction and combining fitted models. Generally, the benefits of multitask learning are more obvious when task overlap is high and datasets are slightly under-sampled [34]. Our results support this principle, as the overall performance increase of multitask network inference for *B. subtilis* is more pronounced than for *S. cerevisiae*, which datasets sample more heterogeneous conditions. Therefore, to benefit from this approach, defining input datasets that share underlying regulatory mechanisms is essential and user-defined.

A key question here, that requires future work, is the partitioning of data into separate datasets. Here we use the boundaries afforded by previous study designs: we use data from two platforms and two strains for *B. subtilis* (a fairly natural boundary) and the separation between studies by different groups (again using different technologies) in yeast. We choose these partitions to illustrate robustness to the more common sources of batch effect in meta-analysis. In the future, we expect that multitask methods in this domain will integrate dataset partition estimation (which data go in which bucket) with network inference. Such methods would ideally be able to estimate task similarity, taking into account principles of regulatory biology, and apply a weighted approach to information sharing. In addition, a key avenue for future work will be to adapt this method to multi-species studies. Examples multitask settings of high biological and biomedical interest include joint inferences that include model system and organisms of primary interest (for example data-set that include mouse and human data collected for similar cell types in similar conditions). These results (and previous work on many fronts [7, 25, 69]) suggest that this method would perform well in this setting. Nevertheless, because of the increasing practice of data sharing in Biology, we speculate that cross-study inference methods will be largely valuable in the near future, being able to learn more robust and generalizable hypotheses and concepts. Although we present this method as an alternative to batch correction, we should point out that there are many uses to batch correction that fall outside of the scope of network inference, and our results do not lessen the applicability of batch correction methods to these many tasks. There is still great value in properly balancing experimental designs when possible to allow for the estimation of specific gene- and condition-wise batch effects. Experiments where we interact MTL learning with properly balanced designs and quality batch correction are not provided here, but would be superior. Thus, the results here should be strictly interpreted in the context of network inference, pathway inference, and modeling interactions.

Methods

Expression data selection, preprocessing and batch-correction

For *B. subtilis*, we downloaded normalized expression datasets from the previously published network study by Arrieta-Ortiz *et al* [17]. Both datasets are available at GEO, *B. subtilis 1* with accession number GSE67023 [17] and *B. subtilis 2* with accession number GSE27219 [52]. For yeast, we downloaded expression datasets from the SPELL database, where hundreds of re-processed gene expression data is available for this organism. In particular, we selected three datasets from separate studies based on the number of samples, within-dataset condition diversity, and cross-dataset condition overlap (such as nutrient-limited stress). *S. cerevisiae 1* and *S. cerevisiae 2* are also available at GEO at accession numbers GSE11452 [54] and GSE9376 [55]. *S. cerevisiae 3* does not have a GEO accession number, and was collected in a custom spotted microarray [56]. For network inference, we only kept genes present in all datasets, resulting in 3780 and 4614 genes for *B. subtilis* and for yeast respectively. In order to join merge, for comparison, we consider each dataset to be a separate batch, since they were generated in different labs as part of separate studies, and applied ComBat for batch-correction using default parameters and no reference to experimental designs [64].

Building priors from chromatin accessibility

ATAC-seq data download, processing, and peak calling

We downloaded chromatin accessibility data *S. cerevisiae* from the European Nucleotide Archive (PRJNA276699) [70, 71]. Reads were mapped to the sacCer3 genome (iGenomes, UCSC) using bowtie2 [72] with the options –very-sensitive –maxins 2000. Reads with low mapping quality (MAPQ < 30), or that mapped to mitochondrial DNA were removed. Duplicates were removed using Picard. Reads mapping the forward strand were offset by +4 bp, and reads mapping to the reverse strand -4 bp. Accessible regions were called using MACS2 [73] with the options –qvalue 0.01 –gsize 12100000 –nomodel –shift 20 –extsize 40. We defined the union of peaks called in any the ATAC-seq samples as the set of putative regulatory regions.

Motifs download, assignment to target genes, and prior generation

We obtained a set of expert-curated motifs for *S. cerevisiae* containing position frequency matrices for yeast transcription factors from The Yeast Transcription Factor Specificity Compendium motifs (YeTFaSCo) [74]. Then, we scanned the whole yeast genome for occurrences of motifs using FIMO with p-value cutoff 1e-4 [59], and kept motifs that intersected with putative regulatory regions. Each motif was then assigned to the gene with closest downstream transcription start site. Gene annotations were obtained from the Saccharomyces Genome Database (SGD) [75]. A list of putative regulators was downloaded from the YeastMine database [61], and then generated a targets-by-regulators matrix (prior) where entries are the count of motifs for a particular regulator assigned to each gene. Finally, we multiplied entries for repressors by -1.

Network inference

We approach network inference by modeling gene expression as a weighted sum of the activities of transcription factors [17, 36]. Our goal is to learn these weights from gene expression data as accurately as possible. In this section, we explain our core model of gene regulation, and of transcription factor activities, and state our assumptions. We also describe how we extend our framework to support learning of multiple networks simultaneously, and integration of prior knowledge on network structure. Finally, we explain how we rank predicted interactions which is used to evaluate the ability of these methods to recover the known underlying network.

Core model

We model the expression of a gene i at condition j, $X_{i,j}$, as the weighted sum of the activities of each transcription factor k at condition j, $A_{k,j}$ [17, 43]. Note that although several methods use transcription factor expression as an approximation for its activity, we explicitly estimate these values from expression data and a set of a prior known interactions. Strength and direction (activation or repression) of a regulatory interaction between transcription factor k and gene i is represented by i, k. At steady state, we assume:

$$X_{i,j} = \sum_{k \in TFs} w_{i,k} \hat{A}_{k,j} \qquad (1)$$

For time-series, we reason that there is a delay τ between transcription factor activities and resulting changes in target gene expression [43]. Given expression of a gene i in time t_n, X_{i,t_n}, and activity of transcription factor k at time $t_{n-\tau}$, $A_{k,t_{n-\tau}}$, we assume:

$$X_{i,t_n} = \sum_{k \in TFs} w_{i,k} \hat{A}_{k,t_{n-\tau}} \tag{2}$$

If time $tn - \tau$ is not available in the expression data, linear interpolation is used to fit $A_{k,t_{n-\tau}}$.

Finally, because we expect each gene to be regulated by only a few transcription factors, we seek a sparse solution for w. That is, a solution in which most entries in w are zero.

Estimating transcription factor activities (TFA)

We use the expression of known targets of a transcription factor to estimate its activity. From a set of prior interactions, we build a connectivity matrix P, where entries represent known activation, $P_{i,k} = 1$, or repression, $P_{i,k} = -1$, of gene i by transcription factor k. If no known interaction, $P_{i,k} = 0$. We assume that the expression of a gene can be written as a linear combination of the activities of its prior known regulators [17].

$$X_{i,j} = \sum_{p \in TFs} P_{i,k} A_{k,j} \tag{3}$$

In case of time-series experiments, we use the expression of genes at time $t_{n+\tau/2}$, $X_{i,t_{n+\tau/2}}$, to inform the activities at time t_n, A_n. Note that for estimating activities, the time delay used is $\tau/2$. Again, linear interpolation is used to estimate $X_{i,t_{n+\tau/2}}$ if gene expression at $t_{n+\tau/2}$ was not measured experimentally [17].

$$X_{i,t_{n+\tau/2}} = \sum_{p \in TFs} P_{i,k} A_{k,t_n} \tag{4}$$

In matrix form, both time-series and steady-state equations can be written as $X = PA$. Since there are more target genes than regulators $i > p$, this is an over-determined system, and thus has no solution, we approximate A by finding $\hat{A}$ that minimizes $||P\hat{A} - X||_2^2$. The solution is given by $\hat{A} = P^*X$, where $P^* = (P^TP)^{-1}P^T$, the pseudo-inverse of P. Finally, for transcription factors with no targets in P, we use the measured expression values as proxy for the activities.

Learning regression parameters

Given gene expression and activity estimates, the next step is to define a set of regulatory hypotheses for the observed changes in gene expression. For each gene, we find a sparse solution for the regression coefficients where nonzero values indicate the transcription factors that better explain the changes observed in gene expression. In this section, we explain how we learn these parameters from a single dataset (single-task learning) and from multiple (multitask learning).

Single-task learning using lasso regression (l_1)

The lasso (least absolute selection and shrinkage operator) is a method that performs both shrinkage of the regression coefficients and model selection [49]. That is, it shrinks regression coefficients towards zero, while setting some of them to exactly zero. It does so by adding a penalty on the sum of the absolute values of the estimated regression coefficients. Let $\hat{A}$ be the activities matrix, X_i the expression values for gene i, and w the vector of coefficients, lasso estimates are given by:

$$\arg\min_{w} \frac{1}{2n} ||X_i - \hat{A}^T w||_2^2 + \lambda ||w||_1 \qquad (5)$$

Where $||w||_1 = \sum_k |w_k|$. When minimizing the above function, we seek a good fit while subject to a “budget” on the regression coefficients. The hyper-parameter λ controls how much weight to put on the l_1 penalty. The lasso became very popular in the last decade, because it reduces overfitting and automatically performs variable selection. We choose the lasso as a single-task baseline because it is equivalent to the S matrix in the multitask case (see below), but with independent choice of sparsity parameter for each dataset.

Multitask learning using sparse block-sparse regression ($l_1/l_1 + l_1/l_\infty$)

We extend our core model to the multiple linear regression setting to enable simultaneous parameter estimation. Here we represent regression parameters for a single gene i as a matrix W, where rows are transcription factors k and columns are networks (or datasets) d. We seek to learn the support $Supp(W)$, where nonzero entries $W_{k,d}$ represent a regulatory interaction between transcription factor k and gene i for network from dataset d.

$$X_{i,j}^{(d)} = \sum_{k \in TFs} W_{k,d} \hat{A}_{k,j}^{(d)} \tag{6}$$

For a given gene i, we could assume that the same regulatory network underlies the expression data in all datasets d. That is, rows in W are either completely non-zero or zero. Since a different set of experiments may have different regulatory patterns, this could be a very strong assumption. A more realistic scenario would be that for each gene i, certain regulators are relevant to regulatory models for all datasets d, while others may be selected independently by each model d. Thus, some rows in the parameter matrix W are entirely nonzero or zero, while others do not follow any particular rule. In this scenario, the main challenge is that a single structural constraint such as row-sparsity does not capture the structure of the parameter matrix W. For these problems, a solution is to model the parameter matrix as the combination of structurally constrained parameters [76].

As proposed by Jalali et al. [32], we learn the regression coefficients by decomposing W into B and S, that encode similarities and differences between regulatory models respectively. This representation combines a block-regularization penalty on B enforcing row-sparsity $||B||_{1,\infty} = \sum_k ||B_k||_\infty$, where $||B_k||_\infty := \max_d |B_{k,d}|$ (as the one from the previous section), and an elementwise penalty on S allowing for deviation across regulatory models for each dataset $||S||_{1,1} = \sum_{k,d} |S_{k,d}|$. The goal is to leverage any parameter overlap between models through B, while accommodating the differences through S. We obtain an estimate for $\hat{W}$ by solving the following optimization problem:

$$\arg\min_{S,B} \frac{1}{2n} \sum_d ||X_i^{(d)} - \hat{A}^{(d)T}(S_{*,d} + B_{*,d})||_2^2 + \lambda_s ||S||_{1,1} + \lambda_b ||B||_{1,\infty} \tag{7}$$

$$output : \hat{W} = \hat{B} + \hat{S}$$

Incorporating prior knowledge using the adaptive lasso

We incorporate prior knowledge by differential shrinkage of regression parameters in S through the adaptive lasso [33]. We choose to apply this only to the S component, because we wanted to allow the user to input different priors for each dataset if so desired. Intuitively, we penalize less interactions present in the prior network. Let Φ be a matrix of regulators k by datasets d, such that entries $\Phi_{k,d}$ are inversely proportional to our prior confidence on the interaction between regulator k and gene i for dataset d. We then optimize the following objective:

$$\arg\min_{S,B} \frac{1}{2n} \sum_d ||X_i^{(d)} - \hat{A}^{(d)T}(S_{*,d} + B_{*,d})||_2^2 + \lambda_s \sum_{k,d} |\Phi_{k,d} S_{k,d}| + \lambda_b ||B||_{1,\infty} \tag{8}$$

$$output : \hat{W} = \hat{B} + \hat{S}$$

We implement this by scaling λ_s by Φ, then the penalty applied to $S_{k,d}$ becomes $\Phi_{k,d}\lambda_s$. In the extreme $\Phi_{k,d} = 0$, the regulator k is not penalized and will be necessarily included in the final model for dataset d. In practice, the algorithm accepts an input prior weight $\rho \geq 1$ that is used to generate the matrix Φ. We apply the complexity-penalty reduction afforded by $\Phi_{k,d}$ to $\hat{S}$ and not $\hat{B}$ as this choice penalizes unique terms, creating the correct behavior of encouraging model differences that are in accord with orthogonal data as expressed in the network-prior. This choice is also in accord with the interpretation of the prior as valid in one, but not necessarily all, conditions. If regulator k is in the prior for dataset d, then $\Phi_{k,d} = 1/\rho$, otherwise $\Phi_{k,d} = 1$. Finally, we rescale $\Phi_{*,d}$ to sum to the number of predictors k. Note that each network model accepts its own set of priors.

Model selection

As proposed by Jalali et al. [32], for MTL, we set $\lambda_b = c\sqrt{\frac{d \log p}{n}}$, with n being the number of samples, d being the number of datasets, and search for c in the logarithmic interval [0.01, 10]. For each λ_b, we look for λ_s that satisfy $\frac{1}{2} < \frac{\lambda_s}{\lambda_b} < 1$. We choose the optimal combination (λ_s, λ_b) that minimizes the extended Bayesian information criterion (EBIC) [77], here defined as:

$$EBIC = \frac{1}{d}\sum_{d} n_d \ln \frac{1}{n_d}||X_i^{(d)} - \hat{A}^{(d)T} W_{*,d}||_2^2 + 2k_d \ln n_d + 2\gamma \ln \binom{p_d}{k_d} \qquad (9)$$

with k_d being the number of nonzero predictors in W for model d, and $0 < \gamma < 1$. Note that for $\gamma = 0$, we recover the original BIC. Whereas for $\gamma > 0$, the EBIC scales with the predictor space k making it particularly appropriate for scenarios where $p >> n$, often encountered in biological network inference projects. In this study, we set $\gamma = 1$. For STL, we use the same EBIC measure, but we calculate it for each dataset separately. Importantly, model selection using EBIC is significantly faster than when compared to re-sampling approaches, such as cross-validation or stability selection [78]. Cross-validation, for example, was previously reported as an impediment for multitask learning in large-scale network inference due computational feasibility [29].

Implementation

We implemented the MTL objective function using cyclical coordinate descent with covariance updates. That is, at each iteration of the algorithm we cycle through the predictors (coordinates), and minimize the objective at each predictor k while keeping the others fixed. Briefly, for a given (λ_s, λ_b), we update entries in S and B respectively, while keeping other values in these matrices unchanged, for several iterations until convergence. First, we update values in S by:

$$\hat{S}_{k,d} = \arg\min_{S_{k,d}} \frac{1}{2} ||R_k^{(d)} - S_{k,d} A_k^{(d)}||_2^2 + \lambda_s \sum_k |S_{*,d}|, \forall k, d \tag{10}$$

with $R_k^{(d)} = X_i^{(d)} - \sum_{l \neq k} (S_{l,d} + B_{l,d}) A_l^{(d)} - \sum_k B_{k,d} A_{k,d}$, being the partial residual vector. Intuitively, we remove effect of the previous coefficient value for $S_{k,d}$, while keeping $B_{k,d}$ unchanged and measure how it changes the residuals. This represents a measure of how important that feature is to the prediction, and contributes to the decision of whether a feature is pushed towards zero or not by the lasso penalty. For $\lambda_s = 0$, we can find the least squares update, $\alpha_{k,d} = \langle R_k^{(d)}, A_k^{(d)} \rangle$, and re-write as $\alpha_{k,d} = \langle A_k^{(d)}, X_i^{(d)} \rangle - \sum_{l \neq k} (S_{l,d} + B_{l,d}) \langle A_l^{(d)}, A_k^{(d)} \rangle - B_{k,d} \langle A_k^{(d)}, A_k^{(d)} \rangle$. This formulation can be optimized much quicker using the covariance updates explained below.

Then, we update $\hat{B}_k$, which represents an entire row in B, by:

$$\hat{B}_k = \arg\min_{B_k} \frac{1}{2} \sum_d ||R_k^{(d)} - B_{k,d} A_k^{(d)}||_2^2 + \lambda_b ||B_k||_\infty, \forall k \tag{11}$$

with $R^{(d)} = X_i^{(d)} - \sum_{l \neq k} (S_{l,d} + B_{l,d}) A_l^{(d)} - \sum_k S_{k,d} A_k^{(d)}$, being the partial residual vector for this case. In this case, we keep the value $S_{k,d}$ unchanged, and set $B_{k,d}$ to zero. Similarly, we remove effects from previous $B_{k,d}$ and evaluate how this feature is for the prediction; this then contributes to the decision of whether this entire row is sent to zero by the infinity norm penalty. For $\lambda_b = 0$, we can find the least squares update, $\alpha_{k,d} = \langle R^{(d)}, A_k^{(d)} \rangle$, which can be re-written as $\alpha_{k,d} = \langle A_k^{(d)}, X_i^{(d)} \rangle - \sum_{l \neq k} (S_{l,d} + B_{l,d}) \langle A_l^{(d)}, A_k^{(d)} \rangle - S_{k,d} \langle A_k^{(d)}, A_k^{(d)} \rangle$. Finally, we apply soft-thresholding to penalize the least-squares updates.

Using these formulations for the updates, we can use the idea of covariance updates [50, 79], where the cross-products $A^T A$ and $A^T X$ are stored in separate matrices and reused at every iteration. Because these cross-products correspond to over 95% of computation time, this trick decreases runtime significantly. To further decrease runtime, we also employ warm starts when searching for optimal penalty values (λ_s, λ_b) [79]. Additionally, since we infer regulators for each gene separately, we can parallelize calculations by gene.

Estimating prediction confidence scores

For each predicted interaction we compute a confidence score that represents how well a predictor explains the expression data, and a measure of prediction stability. As previously proposed [17, 43], we calculate confidence scores for each interaction by:

$$c_{k,i} = 1 - \frac{\sigma^2_{full\ model\ for\ x_i}}{\sigma^2_{model\ for\ x_i\ without\ predictor\ k}} \tag{12}$$

where σ^2 equals the variance of the residuals for the models, with and without predictor k. The score $c_{k,i}$ is proportional to how much removing regulator k from gene i set of predictors decreases model fit. To measure stability, we perform the inference across multiple bootstraps of the expression data (we used 20 bootstraps for both *B. subtilis* and yeast), rank-average the interactions across all bootstraps [16, 43], and re-scale the ranking between 0 and 1 to output a final ranked list of regulatory hypotheses.

References

1. Margolin AA, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Dalla Favera R, et al. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC bioinformatics. 2006;7(1):S7.

2. Petralia F, Wang P, Yang J, Tu Z. Integrative random forest for gene regulatory network inference. Bioinformatics. 2015;31(12):i197–i205.

3. Bonneau R, Reiss DJ, Shannon P, Facciotti M, Hood L, Baliga NS, et al. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. Genome biology. 2006;7(5):R36.

4. Yosef N, Shalek AK, Gaublomme JT, Jin H, Lee Y, Awasthi A, et al. Dynamic regulatory network controlling TH17 cell differentiation. Nature. 2013;496(7446):461.

5. Ciofani M, Madar A, Galan C, Sellars M, Mace K, Pauli F, et al. A validated regulatory network for Th17 cell specification. Cell. 2012;151(2):289–303.

6. Rung J, Brazma A. Reuse of public genome-wide gene expression data. Nature reviews Genetics. 2013;14(2):89.

7. Koch C, Konieczka J, Delorey T, Lyons A, Socha A, Davis K, et al. Inference and Evolutionary Analysis of Genome-Scale Regulatory Networks in Large Phylogenies. Cell systems. 2017;4(5):543–558.

8. Muir P, Li S, Lou S, Wang D, Spakowicz DJ, Salichos L, et al. The real cost of sequencing: scaling computation to keep pace with data generation. Genome biology. 2016;17(1):53.

9. Marx V. Biology: The big challenges of big data. Nature. 2013;498(7453):255–260.

10. Leek JT, Scharpf RB, Bravo HC, Simcha D, Langmead B, Johnson WE, et al. Tackling the widespread and critical impact of batch effects in high-throughput data. Nature reviews Genetics. 2010;11(10).

11. Nayfach S, Pollard KS. Toward accurate and quantitative comparative metagenomics. Cell. 2016;166(5):1103–1116.

12. Pritchard CC, Cheng HH, Tewari M. MicroRNA profiling: approaches and considerations. Nature reviews Genetics. 2012;13(5):358.

13. Tung PY, Blischak JD, Hsiao CJ, Knowles DA, Burnett JE, Pritchard JK, et al. Batch effects and the effective design of single-cell gene expression studies. Scientific reports. 2017;7:39921.

14. Auer PL, Doerge R. Statistical design and analysis of RNA sequencing data. Genetics. 2010;185(2):405–416.

15. Nygaard V, Rødland EA, Hovig E. Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. Biostatistics. 2016;17(1):29–39.

16. Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, et al. Wisdom of crowds for robust gene network inference. Nature Methods. 2012;9(8):796–804.

17. Arrieta-Ortiz ML, Hafemeister C, Bate AR, Chu T, Greenfield A, Shuster B, et al. An experimentally supported model of the *Bacillus subtilis* global transcriptional regulatory network. Molecular Systems Biology. 2015;11(11):839.

18. Dietterich TG, et al. Ensemble methods in machine learning. Multiple classifier systems. 2000;1857:1–15.

19. Papp B, Oliver S. Genome-wide analysis of the context-dependence of regulatory networks. Genome biology. 2005;6(2):206.

20. Harbison CT, Gordon DB, Lee TI, Rinaldi NJ, Macisaac KD, Danford TW, et al. Transcriptional regulatory code of a eukaryotic genome. Nature. 2004;431(7004):99–104.

21. Siahpirani AF, Roy S. A prior-based integrative framework for functional transcriptional regulatory network inference. Nucleic acids research. 2017;45(4):e21–e21.

22. Buenrostro JD, Giresi PG, Zaba LC, Chang HY, Greenleaf WJ. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. Nature methods. 2013;10(12):1213.

23. Boyle AP, Davis S, Shulha HP, Meltzer P, Margulies EH, Weng Z, et al. High-resolution mapping and characterization of open chromatin across the genome. Cell. 2008;132(2):311–322.

24. Johnson DS, Mortazavi A, Myers RM, Wold B. Genome-wide mapping of in vivo protein-DNA interactions. Science. 2007;316(5830):1497–1502.

25. Lam KY, Westrick ZM, Müller CL, Christiaen L, Bonneau R. Fused regression for multi-source gene regulatory network inference. PLoS computational biology. 2016;12(12):e1005157.

26. Omranian N, Eloundou-Mbebi JM, Mueller-Roeber B, Nikoloski Z. Gene regulatory network inference using fused LASSO on multiple data sets. Scientific reports. 2016;6:20533.

27. Jain S, Gitter A, Bar-Joseph Z. Multitask learning of signaling and regulatory networks with application to studying human response to flu. PLoS computational biology. 2014;10(12):e1003943.

28. Wang Y, Joshi T, Zhang XS, Xu D, Chen L. Inferring gene regulatory networks from multiple microarray datasets. Bioinformatics. 2006;22(19):2413–2420.

29. Chasman D, Walters KB, Lopes TJ, Eisfeld AJ, Kawaoka Y, Roy S. Integrating Transcriptomic and Proteomic Data Using Predictive Regulatory Network Models of Host Response to Pathogens. PLoS computational biology. 2016;12(7):e1005013.

30. Gupta R, Stincone A, Antczak P, Durant S, Bicknell R, Bikfalvi A, et al. A computational framework for gene regulatory network inference that combines multiple methods and datasets. BMC systems biology. 2011;5(1):52.

31. Qin J, Hu Y, Xu F, Yalamanchili HK, Wang J. Inferring gene regulatory networks by integrating ChIP-seq/chip and transcriptome data via LASSO-type regularization methods. Methods. 2014;67(3):294–303.

32. Jalali A, Sanghavi S, Ruan C, Ravikumar PK. A dirty model for multi-task learning. In: Advances in Neural Information Processing Systems; 2010. p. 964–972.

33. Zou H. The adaptive lasso and its oracle properties. Journal of the American statistical association. 2006;101(476):1418–1429.

34. Caruana R. Multitask learning. In: Learning to learn. Springer; 1998. p. 95–133.

35. Chen X, Xuan J, Wang C, Shajahan AN, Riggins RB, Clarke R. Reconstruction of transcriptional regulatory networks by stability-based network component analysis. IEEE/ACM transactions on computational biology and bioinformatics. 2013;10(6):1347–1358.

36. Fu Y, Jarboe LR, Dickerson JA. Reconstructing genome-wide regulatory network of *E. coli* using transcriptome data and predicted transcription factor activities. BMC bioinformatics. 2011;12(1):233.

37. Dai Z, Iqbal M, Lawrence ND, Rattray M. Efficient inference for sparse latent variable models of transcriptional regulation. Bioinformatics. 2017;33(23):3776–3783.

38. Liao JC, Boscolo R, Yang YL, Tran LM, Sabatti C, Roychowdhury VP. Network component analysis: reconstruction of regulatory signals in biological systems. Proceedings of the National Academy of Sciences. 2003;100(26):15522–15527.

39. Sanguinetti G, Lawrence ND, Rattray M. Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities. Bioinformatics. 2006;22(22):2775–2781.

40. Filtz TM, Vogel WK, Leid M. Regulation of transcription factor activity by interconnected post-translational modifications. Trends in pharmacological sciences. 2014;35(2):76–85.

41. Ravasi T, Suzuki H, Cannistraci CV, Katayama S, Bajic VB, Tan K, et al. An atlas of combinatorial transcriptional regulation in mouse and man. Cell. 2010;140(5):744–752.

42. Shlyueva D, Stampfel G, Stark A. Transcriptional enhancers: from properties to genome-wide predictions. Nature Reviews Genetics. 2014;15(4):272–286.

43. Greenfield A, Hafemeister C, Bonneau R. Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks. Bioinformatics. 2013;29:1060–1067.

44. Han H, Shim H, Shin D, Shim JE, Ko Y, Shin J, et al. TRRUST: a reference database of human transcriptional regulatory interactions. Scientific reports. 2015;5:11432.

45. Gama-Castro S, Salgado H, Peralta-Gil M, Santos-Zavaleta A, Muniz-Rascado L, Solano-Lira H, et al. RegulonDB version 7.0: transcriptional regulation of Escherichia coli K-12 integrated within genetic sensory response units (Gensor Units). Nucleic acids research. 2010;39(suppl_1):D98–D105.

46. Michna RH, Zhu B, Mäder U, Stülke J. Subti Wiki 2.0—an integrated database for the model organism Bacillus subtilis. Nucleic acids research. 2015;44(D1):D654–D662.

47. Teixeira MC, Monteiro P, Jain P, Tenreiro S, Fernandes AR, Mira NP, et al. The YEASTRACT database: a tool for the analysis of transcription regulatory associations in Saccharomyces cerevisiae. Nucleic acids research. 2006;34(suppl_1):D446–D451.

48. Arnone MI, Davidson EH. The hardwiring of development: organization and function of genomic regulatory systems. Development. 1997;124(10):1851–1864.

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

50. Liu H, Palatucci M, Zhang J. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM; 2009. p. 649–656.

51. Tchourine K, Vogel C, Bonneau R. Explicit Modeling of RNA Stability Improves Large-Scale Inference of Transcription Regulation. bioRxiv. 2017;doi:10.1101/104885.

52. Nicolas P, Mäder U, Dervyn E, Rochat T, Leduc A, Pigeonneau N, et al. Condition-dependent transcriptome reveals high-level regulatory architecture in Bacillus subtilis. Science. 2012;335(6072):1103–1106.

53. Hibbs MA, Hess DC, Myers CL, Huttenhower C, Li K, Troyanskaya OG. Exploring the functional landscape of gene expression: directed search of large microarray compendia. Bioinformatics. 2007;23(20):2692–2699.

54. Knijnenburg TA, Daran JMG, van den Broek MA, Daran-Lapujade PA, de Winde JH, Pronk JT, et al. Combinatorial effects of environmental parameters on transcriptional regulation in Saccharomyces cerevisiae: a quantitative analysis of a compendium of chemostat-based transcriptome data. BMC genomics. 2009;10(1):53.

55. Smith EN, Kruglyak L. Gene–environment interaction in yeast gene expression. PLoS biology. 2008;6(4):e83.

56. Hughes TR, Marton MJ, Jones AR, Roberts CJ, Stoughton R, Armour CD, et al. Functional discovery via a compendium of expression profiles. Cell. 2000;102(1):109–126.

57. Costanzo MC, Engel SR, Wong ED, Lloyd P, Karra K, Chan ET, et al. Saccharomyces genome database provides new regulation data. Nucleic acids research. 2013;42(D1):D717–D725.

58. Weirauch MT, Yang A, Albu M, Cote AG, Montenegro-Montero A, Drewe P, et al. Determination and inference of eukaryotic transcription factor sequence specificity. Cell. 2014;158(6):1431–1443.

59. Grant CE, Bailey TL, Noble WS. FIMO: scanning for occurrences of a given motif. Bioinformatics. 2011;27(7):1017–1018.

60. Karwacz K, Miraldi ER, Pokrovskii M, Madi A, Yosef N, Wortman I, et al. Critical role of IRF1 and BATF in forming chromatin landscape during type 1 regulatory cell differentiation. Nature immunology. 2017;18(4):412.

61. Balakrishnan R, Park J, Karra K, Hitz BC, Binkley G, Hong EL, et al. YeastMine—an integrated data warehouse for Saccharomyces cerevisiae data as a multipurpose tool-kit. Database. 2012;2012.

62. Tsompana M, Buck MJ. Chromatin accessibility: a window into the genome. Epigenetics & chromatin. 2014;7(1):33.

63. Kittler J, Hatef M, Duin RP, Matas J. On combining classifiers. IEEE transactions on pattern analysis and machine intelligence. 1998;20(3):226–239.

64. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics. 2007;8(1):118–127.

65. Müller C, Schillert A, Röthemeier C, Trégouët DA, Proust C, Binder H, et al. Removing Batch Effects from Longitudinal Gene Expression-Quantile Normalization Plus ComBat as Best Approach for Microarray Transcriptome Data. PloS one. 2016;11(6):e0156594.

66. MacNeil LT, Walhout AJ. Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. Genome research. 2011;21(5):645–657.

67. Gitter A, Siegfried Z, Klutstein M, Fornes O, Oliva B, Simon I, et al. Backup in gene regulatory networks explains differences between binding and knockout results. Molecular systems biology. 2009;5(1):276.

68. Hecker M, Lambeck S, Toepfer S, Van Someren E, Guthke R. Gene regulatory network inference: data integration in dynamic models—a review. Biosystems. 2009;96(1):86–103.

69. Waltman P, Kacmarczyk T, Bate AR, Kearns DB, Reiss DJ, Eichenberger P, et al. Multi-species integrative biclustering. Genome biology. 2010;11(9):R96.

70. Schep AN, Buenrostro JD, Denny SK, Schwartz K, Sherlock G, Greenleaf WJ. Structured nucleosome fingerprints enable high-resolution mapping of chromatin architecture within regulatory regions. Genome research. 2015;25(11):1757–1770.

71. Leinonen R, Akhtar R, Birney E, Bower L, Cerdeno-Tárraga A, Cheng Y, et al. The European nucleotide archive. Nucleic acids research. 2010;39(suppl_1):D28–D31.

72. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature methods. 2012;9(4):357–359.

73. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, et al. Model-based analysis of ChIP-Seq (MACS). Genome biology. 2008;9(9):R137.

74. de Boer CG, Hughes TR. YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities. Nucleic acids research. 2011;40(D1):D169–D179.

75. Cherry JM. The Saccharomyces Genome Database: A Tool for Discovery. Cold Spring Harbor Protocols. 2015;2015(12):pdb–top083840.

76. Yang E, Ravikumar PK. Dirty statistical models. In: Advances in Neural Information Processing Systems; 2013. p. 611–619.

77. Chen J, Chen Z. Extended Bayesian information criteria for model selection with large model spaces. Biometrika. 2008;95(3):759–771.

78. Meinshausen N, Bühlmann P. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2010;72(4):417–473.

79. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software. 2010;33(1):1.

Figure Legends

Fig 1: Gene regulatory network inference schematic. (A) Our network inference algorithm takes as input a gene expression matrix, X, and a prior on network structure and outputs regulatory hypotheses of regulator-target interactions. (B) Using priors on network topology and gene expression data, we estimate transcription factor activities (TFA), and subsequently model gene expression as a function of these activities. (C) We use several possible sources of prior information on network topology. (D) Prior information is encoded in a matrix P, where positive and negative entries represent known activation and repression respectively, whereas zeros represent absence of known regulatory interaction. To estimate hidden activities, we consider $X = PA$ (top), where the only unknown is the activities. Of note, a time-delay is implemented for time-series experiments (bottom). (E) Finally, for each gene, we find regulators that influence its expression using regularized linear regression. We either learn these influences, or weights, for each dataset independently, single-task learning (top), or jointly through multi-task learning (bottom).

Fig 2: Representation of the weights matrix for one gene in the multitask setting. We represent model coefficients as a matrix W (predictors by datasets) where nonzero rows represent predictors relevant for all datasets. We decompose the weights into two components, and regularize them differently, using a sparse penalty (l_1/l_1 to S component) to encode a dataset-specific component and a block-sparse penalty (l_1/l_∞ to B component) to encode a conserved one. To illustrate, in this example, non-zero weights are shown on the right side. Note that, in this schematic example, regulators w3 and w7 are shared between all datasets. We also show (bottom) the objective function minimized to estimate S and B on the bottom (for details, see methods).

Fig 3: Multitask learning increase confidence scores of interactions in the gold-standard, improving accuracy of inferred networks. (A) Relationship

between TF activity and target expression in *B. subtilis* 1 (blue) and in B. subtilis 2 (orange), and corresponding STL and MTL inferred confidence scores for an example of an interaction in the *B. subtilis* gold-standard, sigD to lytA. (B) as shown in (A), but for an interaction in the *S. cerevisiae* gold-standard, Msn2 to Hsp104. (C) Inferred confidence scores for interactions in the gold-standard for all datasets. In each plot, dots on the left show scores learned through STL, whereas dots on the right are scores learned through MTL. Lines connect scores associated with the same interaction. (D) Precision-recall curves assessing accuracy of network models inferred for individual *B. subtilis* datasets against a leave-out set of interactions. For simplicity, only one replicate is shown in the curve. Barplot with mean area under precision-recall curve (AUPR) for each method and dataset. Error bars show the standard deviation across 10 splits of the gold-standard into prior and evaluation set. (E) Precision-recall curves assessing accuracy of network models inferred for individual *S. cerevisiae* networks, with the difference that the prior is from an independent source (no splits or replicates).

Fig 4: Multitask learning performance boost outweighs benefits of other data integration methods. Assessment of accuracy of network models learned using three different data integration strategies, data merging and batch correction (STL-BC), ensemble method combining models learned independently (STL-C), and ensemble method combining models learned jointly (MTL-C). (A) Precision-recall curves for *B. subtilis*, again using a leave-out set of interactions. For simplicity, only one replicate is shown in the curve. Barplot with mean area under precision-recall curve (AUPR) for each method. Error bars show the standard deviation across 10 splits of the gold-standard into prior and evaluation set. (B) Precision-recall curves for *S. cerevisiae*, with the difference that the prior is from an independent source (no splits or replicates).

Fig 5: Recovery of prior interactions depends on prior quality and is robust to increasing prior weights. Distribution of number of regulators per target in the *B.*

subtilis prior (A), for the *S. cerevisiae* gold-standard (B), and for the *S. cerevisiae* chromatin accessibility-derived priors (C). (D) Distributions of MTL inferred confidence scores for interactions in the prior for each dataset. Different colors show prior weights used, and represent an amount by which interactions in the prior are favored by model selection when compared to interactions without prior information.

Fig 6: Overlap of edges in inferred networks is higher for *B. subtilis* than for *S. cerevisiae*. Edges overlap across networks inferred using multitask learning for *B. subtilis* (prior weight of 1.0) (A), for *S. cerevisiae* (using the gold-standard as priors) (B), for *S. cerevisiae* (using the chromatin accessibility-derived priors) (C).

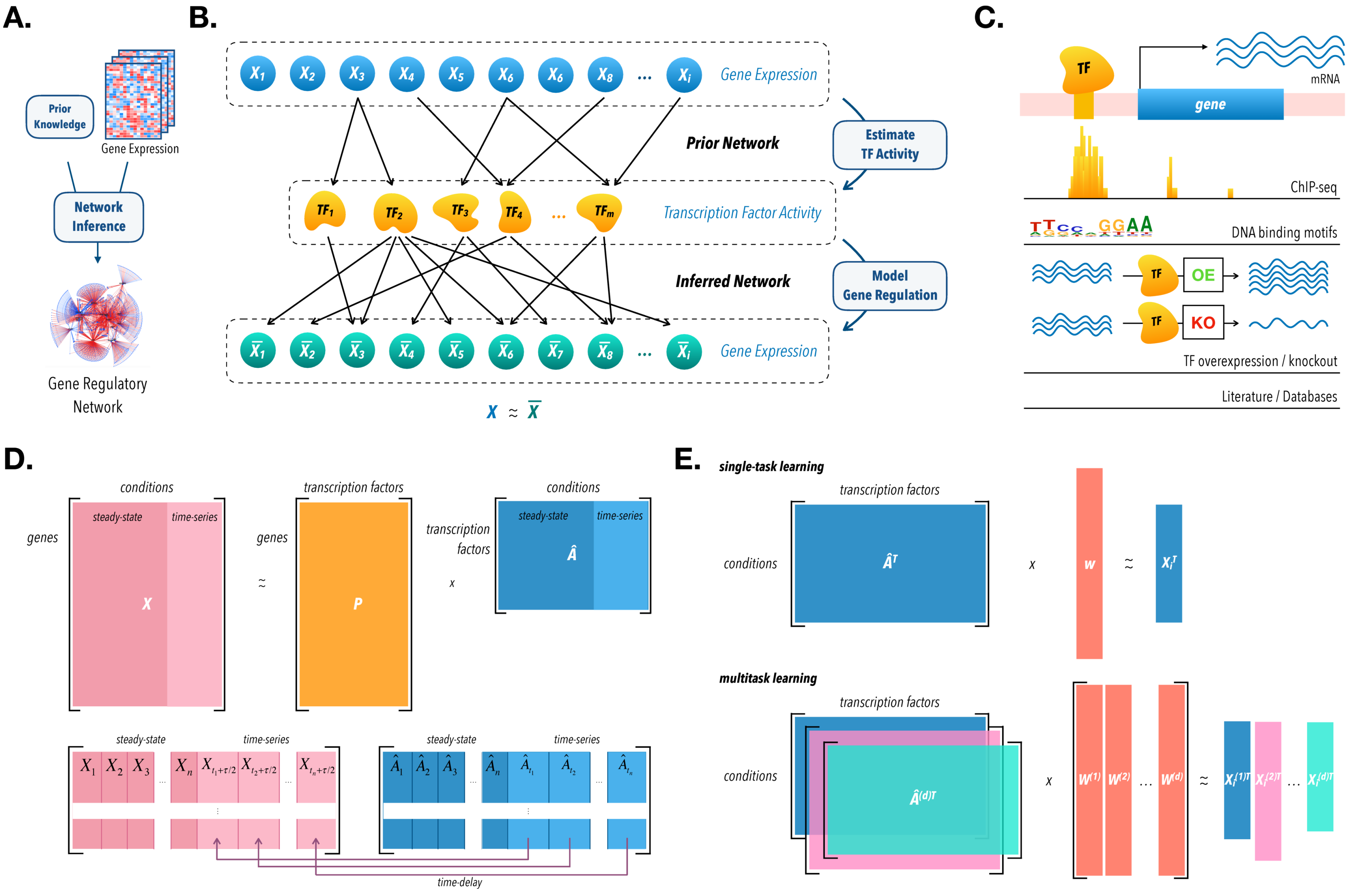
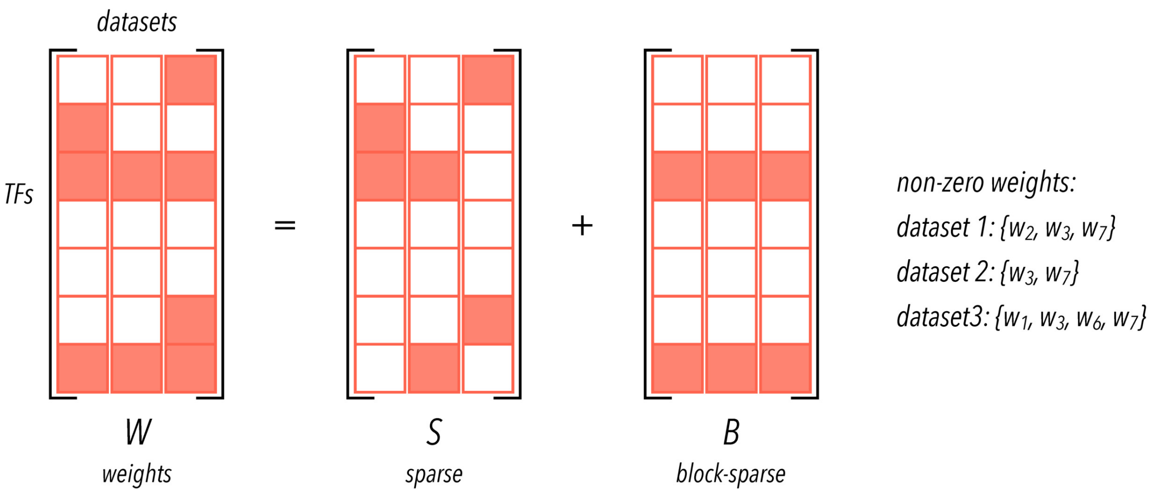


Fig 1. Gene regulatory network inference schematic



$$\arg\min_{S,B} \frac{1}{2n} \sum_{d} ||X_i^{(d)} - \hat{A}^{(d)T}(S_{*,d} + B_{*,d})||_2^2 + \lambda_s \sum_{k,d} |\Phi_{k,d} S_{k,d}| + \lambda_b ||B||_{1,\infty}$$

$$output : \hat{W} = \hat{B} + \hat{S}$$

Fig 2. Representation of the weights matrix for one gene in the multitask setting.

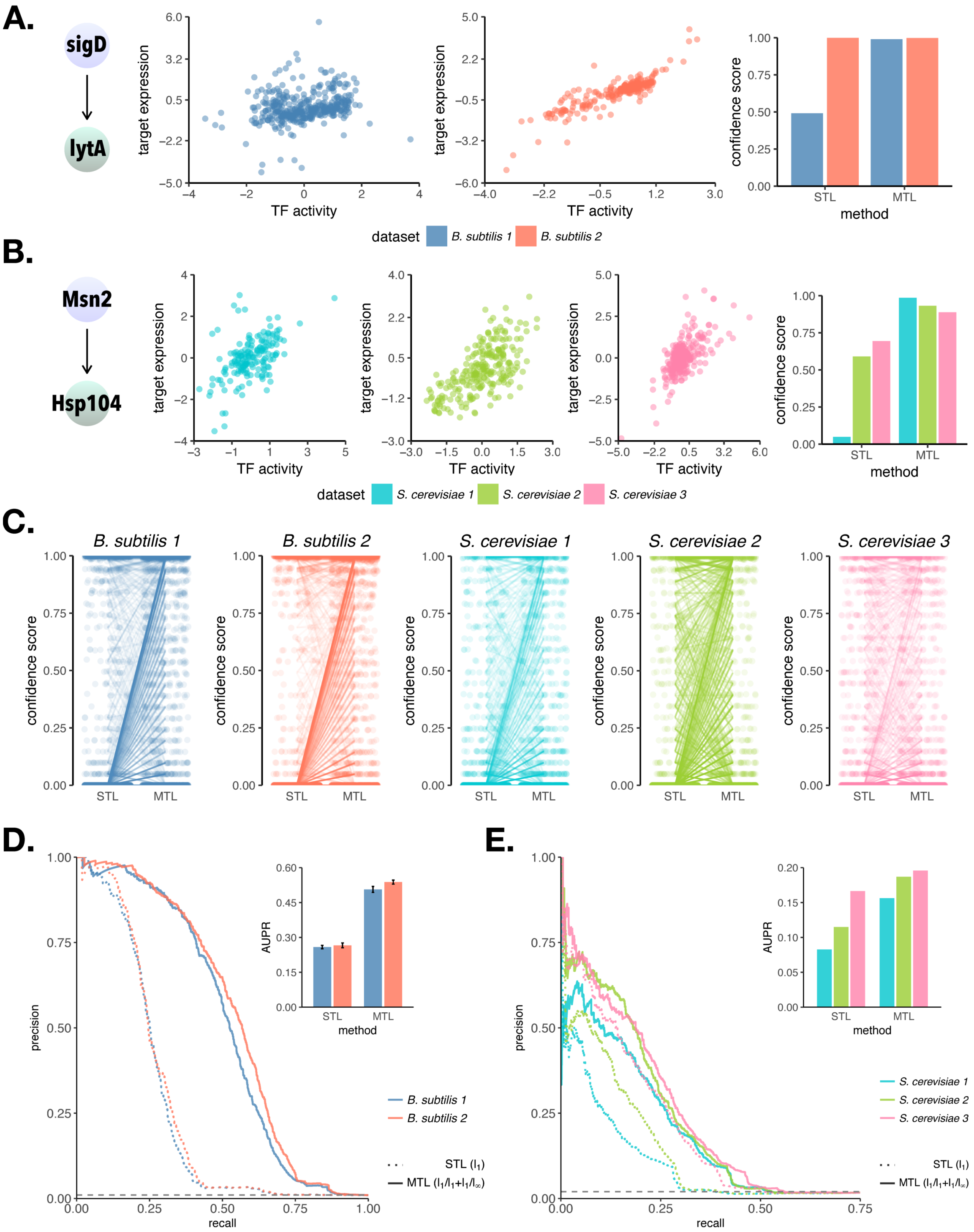


Fig 3. Multitask learning Increase Confidence Scores of Interactions in the Gold-Standard, thus Improving Accuracy of Inferred Networks

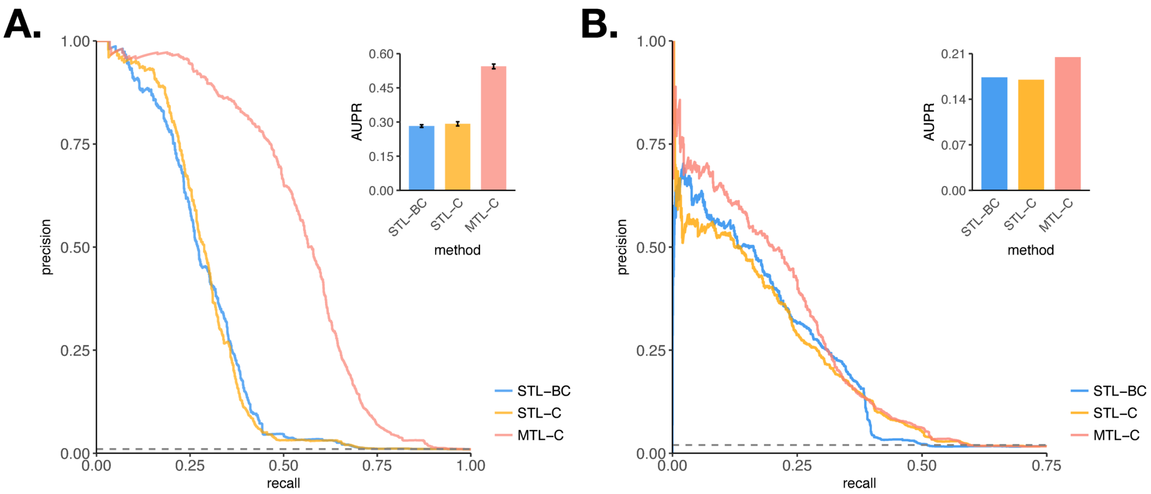


Fig 4. Multitask Learning Boost in Performance Outweights Benefits of Other Data Integration Methods

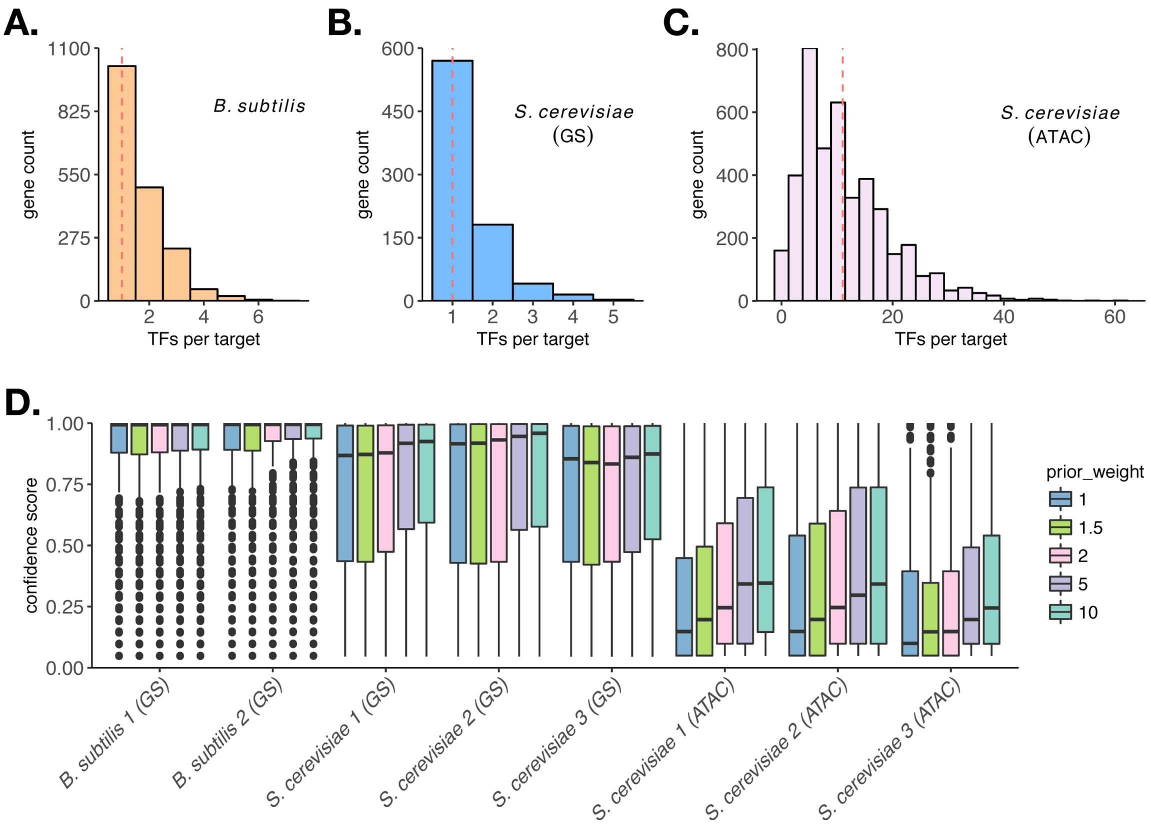


Fig 5. Recovery of Prior Interactions Depends on Prior Quality and is Robust to Increasing Prior Weights

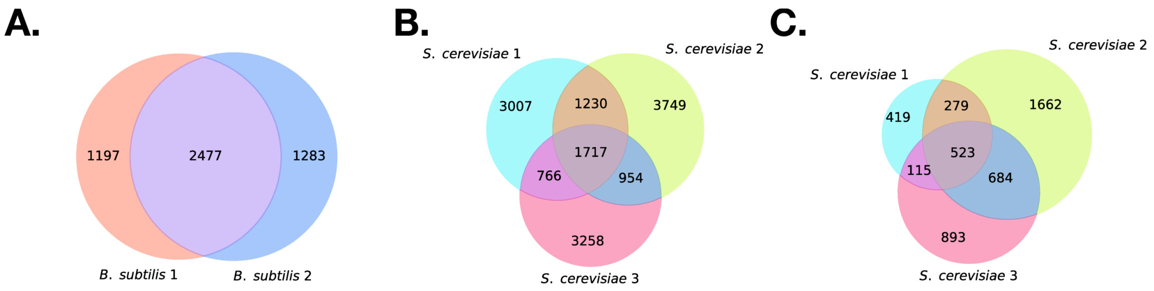


Fig 6. Cross-dataset overlap of inferred edges is higher for *B. subtilis* than for *S. cerevisiae*