Supplementary Materials

Aaron N Brooks David J Reiss

April 18, 2014

Contents

Abstract

Microbes can tailor transcriptional responses to diverse environmental challenges despite having streamlined genomes and a limited number of regulators. Here, we present data-driven models that capture the dynamic interplay of the environment and genome-encoded regulatory programs of two types of prokaryotes: E. coli (a bacterium) and H. salinarum (an archaeon). The models reveal how the genome-wide distributions of cis-acting gene regulatory elements and the conditional influences of transcription factors at each of those elements encode programs for eliciting a wide array of environment-specific responses. We demonstrate how these programs partition transcriptional regulation of genes within regulons and operons to re-organize gene-gene functional associations in each environment. The models capture fitness-relevant co-regulation by different transcriptional control mechanisms acting across the entire genome, to define a generalized, system-level organizing principle for prokaryotic gene regulatory networks that goes well beyond existing paradigms of gene regulation.

1 Online materials

Additional figures, tables, supporting data, and comprehensive model predictions are available at: `http://egrin2.systemsbiology.net`.

2 Supplementary Figures

3 Experimental data

3.1 mRNA expression data

3.1.1 *H. salinarium NRC-1*

A compendium of 1495 transcriptome profiles were collated from a wide array of experiments conducted by our lab that cover dynamic transcriptional responses to varied growth, nutritional, and stress conditions, temperatures, salinities, metal ions, and genetic perturbations (full set of annotations available online). 1159 of these are published (Baliga et al., 2004; Baliga et al., 2002; Bonneau et al., 2007; Facciotti et al., 2010; Facciotti et al., 2007; Kaur et al., 2006; Kaur et al., 2010; Schmid et al., 2011; Schmid et al., 2007; Schmid et al., 2009; Whitehead et al., 2006; Whitehead et al., 2009). 336 are new for this study. Experimental protocols are identical to (Bonneau et al., 2007). These data, including expression levels ($\log_2$ ratios vs. reference samples) and experimental metadata, are available as a tab-delimited spreadsheet.

3.1.1.1 Data normalization

3.1.2 *E. coli K-12 MG1655*

The *E. coli K-12 MG1655*data set was obtained from the DISTILLER website (Lemmens et al., 2009). These data were collated from publicly available microarray data consisting of 3 major microarray databases: Stanford Microarray Database (Demeter et al., 2007), Gene Expression Omnibus (Barrett et al., 2007) and ArrayExpress (Parkinson et al., 2007), The experiments cover a range of conditions, including varying carbon sources, pH, oxygen, metals and temperature.

3.1.2.1 Data normalization

3.2 Data for model validation

3.2.1 *H. salinarium NRC-1*

3.2.1.1 Tiling array transcriptome measurements

3.2.1.2 ChIP-chip transcription factor binding measurements for global regulators

3.2.1.3 Kdp promoter serial truncation measurements

3.2.2 *E. coli K-12 MG1655*

3.2.2.1 Tiling array transcriptome measurements

We measured *E. coli K-12 MG1655*tiling array transcriptome profiles at nine different time points during growth in LB, spanning lag-phase (OD600 = 0.05) to stationary-phase (OD600 = 7.3). RNA samples were prepared as in (Koide et al., 2009). Tiling arrays (Agilent) were custom designed with 60mer probes tiled across both strands of the *E. coli K-12 MG1655* genome using a sliding window of 23bp. Data were quantile-normalized as in (Yoon et al., 2011) and analyzed for condition-specific transcriptional isoforms as in (Koide et al., 2009).

3.2.2.2 PurR/ΔPurR expression data and ChIP-chip transcription factor binding measurements

3.2.2.3 Fitness measurements

3.2.2.4 Effector molecule measurements

3.2.2.5 RegulonDB database of experimentally mapped transcription factor targets

4 Computational methods

4.1 cMonkey: integrated biclustering algorithm, updated for ensemble analysis

4.1.1 Introduction and summary

The cMonkey integrated biclustering algorithm was described and benchmarked in (Reiss et al., 2006). In short, the algorithm computes putatively co-regulated modules of genes over subsets of experimental conditions from gene expression data, constrained by information provided by genome sequence (in the form of de novo identification of conserved cis-regulatory motifs in the gene promoters), and functional association networks. Its defining characteristic is that it integrates all three types of data (expression, sequence and networks) together into an integrated model that is optimized via a stochastic optimization procedure to identify modules that best satisfy all three constraints, simultaneously.

4.1.2 Updates since original publication

For incorporation into the EGRIN 2.0 ensemble analysis, the cMonkey procedure and software was overhauled to improve run-time performance and decrease memory usage. These modifications did not quantifiably affect overall bicluster quality. Changes to the algorithm (and parameters used for EGRIN 2.0 construction) relative to the earlier version described in (Reiss et al., 2006) are as follows:

1. The use of iteratively re-weighted constrained logistic regression to determine gene/condition probabilities for bicluster membership was replaced with a non-parametric cumulative distribution function on gene/condition scores. Since the non-parametric function does not need to be re-weighted, it is significantly faster to compute.

2. Rather than constraining the number of bicluster assignments per gene/condition using a probability distribution, cMonkeynow assigns a fixed number of biclusters to each gene/condition, per run (this is a user-defined parameter, and for this study was set to 2 for genes, and to $k/2$ for conditions, where k is the total number of biclusters in the run, also a user-defined parameter). This modification effectively alters the bicluster optimization from a local (single bicluster) problem with limited cross-bicluster constraints, to a global problem, similar in principle to the widely used k-means clustering algorithm.

3. Since cMonkey uses the updated constraint of (see 2, above) to choose the two "best" biclusters for each gene (and the best $k/2$ biclusters for each condition), there is no sampling as in the prior version. Instead, stochasticity is added, to prevent the optimization from falling into a local minimum, by allowing at most one change in bicluster assignment per gene/condition, per iteration, and by adding a small amount of artificial noise to each gene/condition's bicluster membership weight. This noise occasionally allows moves that decrease a bicluster's total score, and the noise decreases to near zero toward the end of the optimization.

4. The motif search, the most computationally expensive part of the procedure, is limited to run every 100 iterations (for a typical, 2,000 iteration cMonkey run). During the 99 iterations between motif searches, the biclusters are optimized to contain instances of those detected motif(s). We found that this does not impair the ability of cMonkey to detect significant motifs.

5. Finally, as part of the EGRIN 2.0 model construction, only the EMBL STRING (v9) (Szklarczyk et al., 2011) set of predicted gene functional associations, and predicted operons (Price et al., 2005) were integrated (although we note that the software allows other gene association networks to easily be added).

The overall effect of these changes (in addition to other minor modifications and improvements) resulted in an algorithm run-time reduction of about 4-fold. This, in turn, enabled cMonkey to be run numerous times on a modest 8-core compute node (*e.g.*a c1.xlarge Amazon EC2 node) in under six hours per complete run (versus the original cMonkey requiring several days to a week). Practically, the effect of these modifications to the algorithm resulted in a single cMonkey run generating, on average, fewer duplicate biclusters, primarily because each gene is allowed to be a member of only two biclusters. We found that, in general, this increased the overall diversity of regulation (conditional clusterings and corresponding cis-GREs) discovered, per cMonkey run.

4.1.3 Detailed algorithm description

4.1.4 Parameter ranges used for EGRIN 2.0

The additional parameters used for cMonkey, and for MEME (which is used by cMonkeyfor motif detection), are the same as those itemized in (Reiss et al., 2006).

The cMonkey software is available as an open-source R package (Ihaka and Gentleman, 1996). Using this package, the algorithm can be easily applied to nearly any sequenced microbial species (given user-supplied expression data). The package automatically downloads and integrates genome and annotation data from various external sources, including RSAtools (van Helden, 2003); MicrobesOnline (Dehal et al., 2010); EMBL STRING (Szklarczyk et al., 2011), NCBI (Edgar et al., 2002), and KEGG (Ogata et al., 1999). Additionally, the package can generate interactive web-based and Cytoscape output (Shannon et al., 2003), allowing users to explore the resulting modules and motifs in the context of external data, software, and databases via the Gaggle (Shannon et al., 2006). Examples of automatically generated output are available at the cMonkey web site. Supplementary R packages with example expression data for organisms including *H. salinarium NRC-1* and *E. coli K-12 MG1655* are also available from the cMonkey website.

4.2 Inferelator: inference of transcriptional regulatory influences

4.2.1 Introduction and summary

4.2.2 Input list of regulatory factors

4.2.3 Updates since original publication

While Inferelator (Bonneau et al., 2006) and its more recent derivatives (Madar et al., 2009) have been successful at accurately inferring causal regulatory influences using gene expression data, and predicting global regulatory dynamics in new experiments, the algorithm as originally published (Bonneau et al., 2006) required refactoring, and additionally had some notable drawbacks, which were remedied in a new implementation. Modifications included:

1. Removal of "pre-grouping" highly correlated regulators into "TF groups" was removed by replacing the L_1 (LASSO) constraint with the elastic-net linear regression constraint. It has been shown that the elastic-net constraint results in highly correlated predictors being grouped as part of the optimization in this case, using only conditions relevant for each bicluster negating the necessity of pre-grouping the predictors (Zou and Hastie, 2005). We note that in all Inferelator runs the elastic net parameter value was 0.8 (*i.e.*, close to the original LASSO L_1 constraint (=1), but with a small amount of L_2).

2. Elimination of "pre-filtering" of regulators for each bicluster based upon high correlation. The procedure now allows the elastic-net to choose among all potential regulators (excluding TF members of the bicluster, which are automatically considered possible regulators, and are removed from the list of candidate predictors prior to applying the elastic net).

3. Capability to up-weight measurements. This was utilized in the EGRIN 2.0 model to up-weight measurements with lower variance (*i.e.*, more tightly co-expressed) among the genes in a bicluster (*i.e.*, standard weighted linear least-squares, $w_i = 1/\sigma_i^2$).

Additional features, such as the ability to up-weight potential regulators to increase their likelihood of being selected and bootstrapping to more robustly select regulator weights are included in the implementation, but were not specifically utilized as part of the currently described EGRIN 2.0 models. The current implementation of Inferelator, which utilizes the elastic net by default is available as an open-source R package.

4.2.4 Detailed algorithm description

4.3 EGRIN 2.0 model construction

4.3.1 Introduction and summary

Procedures to infer a single Environment and Gene Regulatory Influence Network (EGRIN) model from global data using cMonkey and Inferelator were described previously (Bonneau et al., 2007; Bonneau et al., 2006; Reiss et al., 2006). The EGRIN 2.0 modeling procedure updates this process by applying modified cMonkey and Inferelator algorithms (described above) repeatedly to subsets of the available expression data. The subsets were selected semi-randomly, with available biological information constraining the selection procedure by including whole groups of related experiments when one was randomly selected. For *H. salinarum*, we used manually curated metadata about each experiment to group related experiments. For *E. coli* data set, we did not have readily available metadata, so we instead grouped the conditions based upon individual experiments (*e.g.*, time series). EGRIN 2.0 inference methodology is an ensemble learning approach, more specifically a form of bootstrap aggregation (Breiman, 1996), or sub-bagging.

Advantages of sub-bagging include its simplicity (basic model averaging), its capability to reduce variance of the overall model compared to individual runs (Bhlmann and Yu, 2002), and to avoid overfitting (Krogh and Sollich, 1997). The power of an ensemble learning approach results from its ability to average

out errors in the individual models. In the case of GRN inference, this feature helps to overcome artifacts due to both experimental and algorithmic noise. An incorrect classification from a single model instance is identifiable because it re-occurs infrequently in subsequent runs (assuming it is not the result of a systematic error in the algorithm itself).

Sub-bagging of experimental conditions allows the model to up-weight a restricted set of conditions for each individual EGRIN model in the ensemble. This forces each EGRIN to model regulatory behaviors present within a more narrow range of conditions. As a result, individual EGRIN models have the opportunity to discover features (*e.g.*, conditions, genes, GREs) that may distinguish highly related responses or occur in a very limited number of conditions in the data set. To test this assumption, we performed a separate ensemble analysis of 30 EGRIN models that were generated using the entire *H. salinarum* data set of 1,495 conditions. Across these 30 models, we expected to detect 20 instances of the Bat GRE based on its occurrence in EGRIN 2.0 (*i.e.*, motifs similar to GRE #22; Figure E**??**; (Baliga et al., 2001)). Surprisingly, we did not detect a single GRE that matched the Bat GRE (data not shown) when all conditions were included in each run of the model. This is likely because the anoxic conditions under which the TF that binds this GRE (Bat) is active represent only a small portion of the entire data set.

Ensemble-based approaches are being used more frequently in biological data analyses, including random forests (*i.e.*, bags of decision trees) for classifying genetic precursors to cancer (Breiman, 2001) and the recently-published DREAM5 community ensemble of regulatory network predictions (Marbach et al., 2012), which we used as a benchmark in this manuscript to evaluate EGRIN 2.0 predictions. In principle, our approach is similar to the stochastic LeMoNe algorithm (Joshi et al., 2009), which uses Gibbs sampling to learn ensembles of regulatory modules from gene expression data. EGRIN 2.0 is distinguished from LeMoNe and similar algorithms by its ability to predict transcriptional control mechanisms (*i.e.*, GREs) and the conditions in which they operate, both globally and within individual gene promoters.

To construct and mine the EGRIN 2.0 ensemble we utilized additional model aggregation and compilation procedures, including (1) motif clustering (van Dongen and Abreu-Goodger, 2012) and scanning (Bailey and Gribskov, 1998); (2) association network construction and backbone extraction (Serrano et al., 2009); and (3) network community detection (Ahn et al., 2010). These methods were used to identify GREs and their genome-wide locations, gene-gene co-regulatory associations, and corems, respectively. Each of these procedures is described in detail below.

4.3.2 "Ensemble of EGRINs": generation and storage

4.3.3 Clustering of cis-regulatory motifs to identify GREs

Each cMonkey bicluster contains at least one MEME-detected (Bailey and Gribskov, 1998) de novo cis-regulatory motif. These motifs are used by cMonkey to guide bicluster optimization (in addition to other scoring metrics). There were 86,167 and 269,770 motifs detected across the entire ensemble for *E. coli* and *H. salinarum*, respectively. Each motif was represented in the model as a position-specific scoring matrix (PSSM). To determine which of these motifs represented bona fide GREs (as opposed to false positives), we computed pairwise similarities between all motifs using Tomtom (Gupta et al., 2007) (Euclidean distance metric; q-value threshold of 0.01 and overlap of 6 nt) and clustered the most highly similar PSSM pairs using mcl (Van Dongen, 2008). The Tomtom motif similarity p-value threshold and the mcl inflation parameter (I) were selected to (1) maximize the density of edges between PSSMs inside clusters relative to the edges between clusters, and (2) ensure that the mcl jury pruning synopsis was at least 80 (out of 100). Criterion (1) aims to find a clustering that is as inclusive as possible, while minimizing over-clustering, while (2) is a built-in mcl metric that evaluates the quality of the clusters resulting from the user-selected pruning strategy (I). The final parameters were p-value cutoff = 10^{-6} and mcl $I = 4.5$ for *H. salinarum* ensemble and p-value cutoff = 10^{-5} and mcl $I = 1.5$ for the *E. coli* ensemble. We did not filter the motifs by E-value

or other intrinsic motif quality metrics; rather we enforced a cluster size threshold to ensure that GREs were detected consistently. Clusters containing at least 10 PSSMs were considered GREs (representing individual bicluster detection instances). This resulted in 135 GREs for *H. salinarum* (representing 27,991 motif instances, Table E2) and 337 for *E. coli* (representing 12,773 motif instances, Table E3). Finally, we computed a "combined PSSM" for each cluster as the unweighted mean of aligned PSSMs within each cluster (Figure 2E; Figure E**??**).

4.3.4 Genome-wide scanning of motifs to obtain GRE locations

We used scanning to discover GRE locations that were missed by the rigid definition of a promoter in cMonkey (-250 to +50 nucleotides surrounding the translation start site). This procedure was critical for discovering GREs in non-canonical locations, such as internal to operons.. To discover likely GRE locations throughout the genome, we computed how well each GRE matched every position in the genome using MAST (Bailey and Gribskov, 1998). Specifically, we recorded significant matches for every PSSM in each GRE at each genomic location subject to a position p-value threshold of 10^{-5}. This p-value cutoff corresponds to an expectation of discovering 20 sites at random across the genome. For each GRE, we summed the number of significant matches to each of the GREs PSSMs at each genomic position. These counts were used to represent GRE composition in promoters (Figures 2-3, Figures S3-S5). In addition, we used these scanned locations to identify GREs predominately located inside coding regions. Since these GREs may be spurious (*e.g.*, protein sequence motifs) they were flagged, although they were not removed from our global analysis.

4.3.5 Statistical mining of the relationships in the ensemble

Statistical associations between any entity in the EGRIN 2.0 ensemble (*i.e.*, genes, GREs, conditions, TFs; see Figure 1) can be evaluated using the hypergeometric test for statistical enrichment. We use this basic procedure to identify conditions associated with GRE influence, and GREs associated with gene co-regulation, as we describe below.

4.3.6 Identifying corems

4.3.6.1 Gene-gene co-occurrence network

We post-processed the EGRIN 2.0 ensemble to refine the underlying network structure and discover functionally meaningful gene co-regulatory modules present in the model. To do so, we transformed the ensemble of biclusters into a weighted gene-gene association graph G, where the nodes of G are genes and the weight of edges between the nodes is proportional to their frequency of co-occurrence in biclusters:

$$w_{ij} = \frac{|B_i \cap B_j|}{\min(B_i, B_j)}, \tag{1}$$

where w_{ij} is the weight of the edge between genes i and j, B_i is the set of all biclusters containing gene i. The weights were normalized by the minimum number of biclusters containing either gene, rather than by the more typically applied union (which would make the score identical to the Jaccard Index) to avoid penalizing genes that occur infrequently in biclusters. The sum of edge weights for each gene was normalized to one. This gene-gene co-occurrence network represents how often cMonkey discovers co-regulation between every pair of genes in the genome. We note that since this network is derived from biclusters, it is also a reflection of conditional co-expression and predicted cisregulatory motifs.

4.3.6.2 Network backbone extraction

After transforming the ensemble into a normalized graph, we removed edges that were statistically indistinguishable by multiscale backbone extraction (null hypothesis of uniform edge weight distribution given a node of degree k) (Serrano et al., 2009). We retained all edges satisfying the following relation:

$$\alpha_{ij} = 1 - (k-1)\int_0^{w_{ij}} (1-x)^{k-2} dx \leq 0.05, \tag{2}$$

where α_{ij} is the probability that the normalized weight w_{ij} between genes i and j is compatible with the null hypothesis, and k is the degree of gene i. For *H. salinarium NRC-1*, backbone extraction reduced the number of regulatory edges from 1,576,643 to 141,667; in *E. coli K-12 MG1655*the number of edges was reduced from 3,094,954 to 170,723.

4.3.6.3 Network link-community detection

Following backbone extraction, we detected corems by application of a recently described link-community detection algorithm (Ahn et al., 2010). For this algorithm to work on our data set we modified it to accept input of a weighted graph (Kalinka and Tomancak, 2011). We implemented it in C++ for efficiency. The algorithm computes a similarity score between all pairs of edges sharing a common keystone node, k, according to the Tanimoto coefficient:

$$T(e_{ik}, e_{kj}) = \frac{a_i \cdot a_j}{|a_i|^2 + |a_i|^2 + a_i \cdot a_j}, \tag{3}$$

where

$$a_i = w_{ij} + \frac{\delta_{ij}}{k_i} \sum_{l \in n(i)} w_{il}. \tag{4}$$

Here, T is the Tanimoto coefficient, e_{ik} is the edge between gene i and the keystone gene k, and δ_{ij} is the Kroenecker delta. The score reflects the similarity of gene neighborhoods adjacent to two edges sharing a gene, with the score increasing in value as the number and weight of overlapping adjacent edges increases. To transform the Tanimoto coefficient into a distance metric, we compute $1 - T$.

Following scoring, the edges were aggregated by standard hierarchical clustering. The resulting tree is cut at many thresholds to optimize the local weighted density D of the resulting clusters:

$$D = \frac{1}{M\langle w \rangle} \sum_{c \in C} m_c \langle w \rangle_c \left(\frac{m_c - (n_c - 1)}{n_c(n_c-1)/2 - (n_c - 1)} \right), \tag{5}$$

where M is the total number of edges in the entire network, $\langle w \rangle$ is the average weight of edges in the entire network, C is the set of all link communities at a given threshold, m_c is the number of edges in community c, $\langle w \rangle_c$ is the average weight of edges in community c, and n_c is the number of genes in community c. The density scoring metric D had a clear optimum corresponding exactly to the cutoff that would have been chosen had we used the unweighted scoring metric originally described (available online). Only communities with more than two genes were retained.

Since the communities produced by this algorithm are comprised of sets of edges, we defined a corem to include all genes incident to the edges in a community. Because of this definition, each gene can be a member of multiple different corems. In *H. salinarum*, this procedure generated 679 corems ranging in size from 3 to 377 genes, covering 1,363 of the 2,400 genes in the genome, and comprising 56,738 co-regulatory associations. In *E. coli*, we discovered 590 corems, ranging in size from 3 to 153 genes, covering 1,572 of 4,213 genes and 25,976 regulatory edges. See Table E1 for additional statistics. Gene-to-corem and corem-to-gene mappings for the *H. salinarum* and *E. coli* models are available online.

4.4 Functional enrichment estimates for genes in corems

We computed functional enrichment for genes organized into corems using DAVID (Dennis et al., 2003) and the DAVIDQuery R-package (Day and Lisovich, 2010). Enrichments for each corem are available on the web site.

4.5 Conditional co-regulation of genes organized in corems

We defined the conditions in which genes in a corem were co-regulated as the set of experiments in which the genes of a corem are more tightly co-expressed than one would expect at chance. We statistically evaluated tight co-expression using relative standard deviation (RSD $= |\sigma/\mu|$) and resampling. We chose RSD (rather than, for example, standard deviation, σ) to avoid over-weighting conditions in which the mean relative expression is close to zero. The significance of an RSD value for a given condition relative to each corem was estimated by resampling: for a corem with k gene members, and for each condition, c, we computed at least 20,000 RSD values for k randomly sampled expression measurements in c, to determine the likelihood that the observed co-expression has lower RSD than expected by chance (p-value < 0.01). The resampling procedure resulted in condition sets for corems that contained from 1.4% to 85.5% of the conditions in *H. salinarium NRC-1* and 7.9% to 66.6% conditions in *E. coli K-12 MG1655*.

4.6 Conditionality of GRE influence

The upstream promoter regions of most genes contain multiple EGRIN 2.0-predicted GREs (*e.g.*, carA in Figure 2). A key insight of our model is that not all of these sites are equally important for controlling gene expression in all experimental conditions. We refer to changes in the relative influence of GREs across conditions as conditional activity of GRE elements. Although, to be clear, we do not imply that the transcriptional activity at a GRE is attributable to the DNA sequence itself, but rather the TF that binds to that sequence in particular environments. We leveraged the GREs discovered in genes grouped into corems and the conditional co-expression of those groups of genes to predict conditionally active GREs in EGRIN 2.0.

Specifically, to discover active GREs for each corem we combined predictions from (1) genome-wide motif scans (Section 5 above) that predict the GRE locations in an expanded region around each genes promoter in the corem using all of the ensemble predictions (1,000 nt window: -875 nt upstream to 125 nt downstream), and (2) the conditions discovered in biclusters that are most representative of the corem (*i.e.*, containing the largest fraction of genes from the corem, top decile). GREs that occurred frequently inthese biclusters were considered putatively responsible for co-regulating the set of genes in the condition-specific context of the corem (q-value 0.05). Finally, we computed the average distances of all GREs to the start codons of each gene in the list (collapsing sites if they occurred within 25 nt of one another). The precise locations of all GREs for the *H. salinarum* dpp operon-related corems (Figure 3) are listed in Table E8, while the locations for GREs involved in conditional modulation of the PurR regulon (Figure 4) are provided in Table E9.

We represented the active GREs upstream of a gene or within a corem as a pie chart, showing the normalized frequency with which the GREs computed above occurred in biclusters containing that gene. For example, if GREs 1, 2, and 3 occurred in 25, 50, and 200 biclusters containing gene A, the pie chart for gene A would have sectors of area 0.09, 0.18, and 0.73 respectively. For corems, we computed the normalized frequency of GREs for all genes of the corem. For example, if GREs 1, 2, and 3 occurred in promoters of 10, 10, and 20 of the genes of the corem, their areas would be 0.25, 0.25, and 0.5 respectively.

4.7 Detection of conditional operons

Conditional-specific transcriptional isoforms of operons were predicted through corem membership. Specifically, if any of the genes in an operon were found in a corem that did not contain all the other genes of the operon, we predicted that the operon had conditional isoforms. Operon annotations for both *H. salinarum* and *E. coli* were derived from MicrobesOnline. All predicted conditional operons, including the specific break sites and transcriptional isoforms is available on the website. The full list of validated predictions is provided in Table E7.

4.8 Environmental ontology construction and usage

We recorded a rich meta-data set for all 1495 experiments conducted for *H. salinarum*. The meta-data includes a detailed description of each experiment, including, for example: media composition, genetic background, concentration of perturbant, internal reference batch id, person who conducted the experiment, etc. We used this meta-information to classify experiments in an ontological framework, where two experiments can share specific meta-descriptions (*e.g.*, 1e-3 mol/L EDTA), or inherit more general relationships from the ontological structure (*e.g.*, chemical perturbation). We used OBO-edit to construct the ontology. The ontology contained 198 terms organized across three primary branches (environmental state, experimental state, and genetic state). The ontology flat file is available for download and meta-data annotations for every array in the dataset are available online.

We used the ontology to classify enriched environmental features for GREs and corems (Figures 3-4). For corems, we used the set of conditions in which genes in the corem are significantly co-expressed (see 9 above) to compute term enrichment using the ontoCAT R-package. Term enrichment was assessed statistically and reported as q-values using the hypergeometric test with Benjamini-Hochberg correction for multiple hypothesis testing.

5 Model evaluation and validation

5.1 *E. coli* network performance: validation with RegulonDB gold standard

5.1.1 Validation of transcription factor binding sites

We compared the genome-wide locations of predicted GREs (Model Construction: 5 and 6 above) in the *E. coli* ensemble to experimentally mapped TF binding sites from RegulonDB (BindingSiteSet table, filtered for experimental evidence and TFs with ≥ 3 unique binding sites; a total of 88 TFs). We considered a GRE to be a significant match to a TF if a significant fraction (FDR ≤ 0.05) of the predicted non-coding locations of its PSSMs constituents overlapped with the known binding locations for that TF (hypergeometric p-value ≤ 0.01; See GRE definition in Model Construction: 5). In cases where multiple TFs significantly matched a GRE, only the most significant was reported. We also observed several instances where more than one GRE significantly matched the same TF. We were unable to determine whether this was the result of incomplete GRE clustering, ambiguities related to GRE scanning, limitations of the experimental data itself, or, by contrast, a reflection of subtle context-dependent variations in the binding preferences of these TFs. Since we did not observe clustering of GREs that map to the same TF upon re-clustering, we hypothesize that the observations may have biological origins, *i.e.*reflect condition-dependent variations in TF binding preferences that are the result, for example, of co-activator/repressor interaction or small molecule binding. It is interesting to note that TFs with the largest fraction of GRE matches include transcriptional dual regulators like FlhDC and UlaR (*i.e.*, TFs with the ability to act as both activators and repressors). This is consistent with the observation that these TFs have context-dependent binding preferences. The complete set of validations, for both TFs and -factors, is listed in Table E4.

5.1.1.1 Comparison with other module detection algorithms

In addition to the comparisons described above, we also compared the number of RegulonDB TFs detected in the EGRIN 2.0 model to individual cMonkey runs as well as several other algorithms that were computed on subsets of the experimental data (similar to the EGRIN 2.0 ensemble; Figure E2C). We evaluated: (a) k-means clustering, (b) WGCNA, and (c) DISTILLER (Lemmens et al., 2009). For (a) and (b), we computed modules 100 times on random subsets of the *E. coli* expression data set (using 200-250 randomly chosen experiments per run; selection criteria were identical to *E. coli* EGRIN 2.0). We then predicted de novo cis-regulatory GREs in the promoter regions of genes in each module using MEME (MEME parameters were identical to EGRIN 2.0). For (c), we performed the comparison using the original modules generated by (Lemmens et al., 2009). Rather than alter module composition by re-detection, we instead varied MEME parameters applied to the modules 100 times (within the same ranges as those used for EGRIN 2.0). TF-GRE matches were assigned by comparing GREs to RegulonDB TF binding sites, as previously described (Model Evaluation and Validation: 1).

We found that individual cMonkey runs discovered a greater number of RegulonDB binding sites on average than the other methods (41 compared to 30, 25, and 29 for k-means, WGCNA, and DISTILLER, respectively), which is consistent with previous findings (Reiss et al., 2006). Integration of individual cMonkey biclusters into the EGRIN 2.0 ensemble outperformed all individual cMonkey runs. This result is typical of ensemble-based inference approaches, which supports value of ensemble integration as part of the EGRIN 2.0 model.

5.1.2 Comparison with "direct inference" networks from CLR and DREAM5

We subdivided the inferred *E. coli* EGRIN 2.0 GRN into two networks: (1) a GRN derived from Inferelator-predicted transcriptional influences and (2) a GRN derived from TF-matched GREs detected in gene promoters (Model Construction: 5 and 6 above). For (1), TF-gene associations were inferred through TF-bicluster influence (*i.e.*, each gene in a bicluster was assigned to the TFs inferred to regulate the bicluster). TF-gene associations were ranked based upon the number of times that they were observed across the entire EGRIN 2.0 ensemble. The top 100,000 rankings were retained in the final network.

The CLR GRN was computed using default parameters on the same expression data set as EGRIN 2.0 (number of bins = 10 and spline degree = 3). Interactions were sorted based on the CLR score. The top 100,000 interactions were retained in the final network. CLR analysis was performed using MATLAB.

The DREAM5 network was retrieved from (Marbach et al., 2012).

Precision-recall statistics were computed for each of the predicted *E. coli* GRNs using RegulonDB (version 7.2). We used regulatory interactions annotated as having strong experimental evidence (Gama-Castro et al., 2011). The resulting gold-standard network contained 2,427 TF-gene interactions between 155 TFs and 1163 genes. The precision-recall and AUPR statistics were calculated as in (Marbach et al., 2012).

5.2 Validation of conditional operons in tiling array transcriptome measurements

5.3 Global evaluation of fitness correlations

We defined gene modules using regulons (annotated in RegulonDB or RegPrecise) by grouping together genes that were annotated as controlled by a common TF. Using the same community detection procedures that we used to define corems from the EGRIN 2.0 ensemble, we computed gene co-expression modules from the weighted WGCNA adjacency matrix. We compared the distributions of Pearson correlations between relative changes in fitness across pairs of genes within each module, using the one-tailed Kolmogorov-Smirnov test (KS-test). The precision/recall characteristics for each model are contained in Table E5.

References