Lessons from the DREAM2 Challenges
A Community Effort to Assess Biological Network Inference

Gustavo Stolovitzky, Robert J. Prill, and Andrea Califano

*IBM Computational Biology Center, Yorktown Heights, New York, USA
bDepartment of Biomedical Informatics, Columbia University, New York, New York, USA

Regardless of how creative, innovative, and elegant our computational methods, the ultimate proof of an algorithm’s worth is the experimentally validated quality of its predictions. Unfortunately, this truism is hard to reduce to practice. Usually, modelers produce hundreds to hundreds of thousands of predictions, most (if not all) of which go untested. In a best-case scenario, a small subsample of predictions (three to ten usually) is experimentally validated, as a quality control step to attest to the global soundness of the full set of predictions. However, whether this small set is even representative of the global algorithm’s performance is a question usually left unaddressed. Thus, a clear understanding of the strengths and weaknesses of an algorithm most often remains elusive, especially to the experimental biologists who must decide which tool to use to address a specific problem. In this chapter, we describe the first systematic set of challenges posed to the systems biology community in the framework of the DREAM (Dialogue for Reverse Engineering Assessments and Methods) project. These tests, which came to be known as the DREAM2 challenges, consist of data generously donated by participants to the DREAM project and curated in such a way as to become problems of network reconstruction and whose solutions, the actual networks behind the data, were withheld from the participants. The explanation of the resulting five challenges, a global comparison of the submissions, and a discussion of the best performing strategies are the main topics discussed.

Key words: systems biology; network theory; pathway inference; reverse engineering; mathematical modeling; assessment methods; machine learning

You may say I’m a DREAMer, but I’m not the only one . . .

John Lennon

Introduction

The Dialogue for Reverse Engineering Assessments and Methods (DREAM) project had its formal kickoff in the spring of 2006, when a group of systems biologists, from both computational and experimental extraction, convened at the New York Academy of Sciences in Manhattan. What distinguished these scientists was a common passion for understanding biology at a genome-wide level and a common frustration with the increasingly artificial boundaries between the experimental and the computational sciences in modern biological research. In a very insightful meeting1 the group discussed the need for, and timelines of, an effort to “take the pulse” of how well the theoretical reverse-engineering community was objectively doing in reconstructing protein–protein, gene regulatory, signaling, and metabolic networks.2,3 The consensus was that a full-fledged effort to harness and understand the limitations and strengths of the plethora of biological
reverse-engineering methods was going to be a key prerequisite to their increasing value to biology. Indeed, at that time, the field of reverse engineering biological networks was beginning to experience considerable expansion, which generated much confusion about which methods were truly valuable from a practical perspective. Evidence of this trend is shown in Figure 1, where we report the number of publications retrieved from a PubMed search of “Pathway Inference” or “Reverse Engineering.” Figure 1 shows what appears to be an exponential growth, in which citations to these key words have been roughly doubling every two years for the last decade or so.

While such growth in the number of reverse-engineering–related publications was fueled by very innovative and elegant computational methods for network reconstruction—arising from physics, computer science, mathematics, and engineering—the group shared the feeling that, ultimately, an algorithm’s worth was to be found in the experimentally validated quality of its predictions. A key problem is that computational methods can, in the blink of an eye, generate large numbers of predictions, from a few hundred to hundreds of thousands, most (if not all) of which usually go untested. Even worse, and this would be a best case scenario, a very small subsample of predictions—usually three or more, but rarely more than ten—would be validated using sound experimental assays and then presented as valuable criteria for the soundness of the entire set of predictions. Thus, a clear characterization of the relative strengths and weaknesses of the algorithms on an objective basis was usually missing. It should be noted that the same could be said for high-throughput (and even low-throughput) experimental approaches, whose false positive and, equally importantly, false negative rates are rarely considered a requisite for publication. This generated, for instance, more than a few puzzled looks when the first experimentally generated, genome-wide interactomes in yeast showed minimal overlap.

At the spring 2006 meeting, it was agreed that while the obstacles for the creation of a
methodology of reverse engineering assessment would be formidable, the value of such efforts would also be considerable and thus justified. The difficulties in creating well-posed challenges for network reconstruction are multifold, including what type(s) of experimental data to provide for the reconstruction, how these data should be generated, how to ensure appropriate characterization of the ground truth (i.e., the molecular interaction model) supported by these data, and how to objectively evaluate the performance of the challenge participants, among many others. A discussion of these topics was presented in a previous work, and it is unnecessary to further discuss these considerations here. It is sufficient to say that the outcome of the thinking and discussions of the DREAM community culminated in the first set of challenges that was presented to the community in mid-2007: the DREAM2 challenges.

Similar to CASP, the well-known Critical Assessment of Structure Predictions community-wide experiment, the DREAM2 Challenges were based on donated data sets. The identity of the data donors was kept anonymous, and some information pertaining to the data was masked, withheld from public disclosure, or otherwise obfuscated. Depending on the particular challenge, the objective of the DREAM2 challenges was to infer an underlying molecular interaction network, including transcriptional and post-translational interactions, as well as mixed biological circuits of various kinds. In this chapter we describe the five challenges of DREAM2, the community response to these challenges (the submitted reverse engineered networks), and the comparative performance of the different challenge participant groups. We conclude with global lessons extracted from this community-wide exercise.

The DREAM2 Challenges

The BCL6 Transcriptional-Target Challenge

BCL6 is a transcription factor that plays a key role in both normal and pathological B cell physiology. Unpublished data on BCL6 genomic binding sites formed the basis of a challenge in which participants attempted to discriminate functionally validated BCL6 transcriptional targets from “decoy” targets for which evidence indicates no physical and functional control by BCL6.

The Venn diagram of Figure 2 illustrates the procedure for selecting the BCL6 targets and decoy genes for Challenge 1. We took as targets the 53 genes resulting from the intersection of genes identified through ChIP-on-chip as bound to BCL6 (“ChIP-on-chip Positives” set) and genes that responded to BCL6 perturbations (“Functionally validated” set), but that were not identified in the Polo et al. ChIP-on-chip experiment (“Polo et al.” set). One hundred forty seven decoy genes were randomly chosen among the genes that were not in any of these three data sets. (Color is shown in online version.)

FIGURE 2. Venn diagram illustrating the selection the BCL6 targets and decoy genes for Challenge 1. We took as targets the 53 genes resulting from the intersection of genes identified through ChIP-on-chip as bound to BCL6 (“ChIP-on-chip Positives” set) and genes that responded to BCL6 perturbations (“Functionally validated” set), but that were not identified in the Polo et al. ChIP-on-chip experiment (“Polo et al.” set). One hundred forty seven decoy genes were randomly chosen among the genes that were not in any of these three data sets. (Color is shown in online version.)
Intersection of these two datasets provided a set of 56 genes whose promoter regions are bound by BCL6 and that respond to BCL6 perturbations. An independently published dataset of ChIP-on-chip data measured in diffuse large B cell lymphomas\textsuperscript{13} identified a cohort of 437 genes whose promoters bound to BCL6. Only 3 of these 437 genes were also part of the 56 genes, leaving only 53 previously unknown genes that were identified as BCL6 targets with evidence of BCL6 function. An additional set of 147 decoy genes were randomly selected among the genes that were at “the bottom” of the ChIP-on-Chip statistical significance ($P > 0.5$) and that were not identified as candidate BCL6 targets either by the functional analysis or by the previously published ChIP-on-Chip experiment.\textsuperscript{13}

Participants were provided the set of 200 genes and asked to identify the BCL6 targets from the decoys. The information of how many of the 200 genes were actual BCL6 target genes was withheld from the participants. A panel of 336 microarray gene expression profiles for a wide variety of normal, tumor related, and experimentally manipulated human B-cell populations previously published\textsuperscript{12} was also provided to the participants. This panel contained the measurements originally used to identify the 102 genes that responded to the BCL6 perturbations, but this information was not made explicit to the participants. ChIP-on-chip assays were not disclosed to the participants. The Protein–Protein Subnetwork Challenge

The yeast two-hybrid (Y2H) system is a high-throughput method for discovering interactions between pairs of proteins.\textsuperscript{14} The basic idea behind this technology is the activation of a reporter gene by the binding of a transcription factor onto an upstream activating sequence. In the Y2H screening, the transcription factor is divided into two separate fragments: the binding domain (BD) and the activating domain (AD). The BD (the domain responsible for binding the promoter of the reporter gene) is fused to one of our proteins of interest, called the bait. The AD (the domain responsible for activation of transcription) is fused to the other protein of interest, called the prey. The expression of the reporter gene is taken as an indication that the bait captures (interacts) with the prey.

The procedure for the generation of the “gold standard” for this challenge was as follows. In experiments performed in the laboratory of Dr. Marc Vidal at the Dana Farber Cancer Institute, 67 proteins were used as bait (that is, were fused with the BD) and 80 proteins were used as prey (that is, fused to the AD). In independent experiments, each of the $67 \times 80$ assays of protein–protein interactions was performed and replicated nine times. If a pair of proteins was observed to interact in three or more replicas, the pair was marked as a putative protein–protein interaction. If any of those pairs had been reported to interact previously in the literature, then the pair was removed from our list of interacting proteins (because we desired that participants only predict interactions that were not previously reported). This process yielded 48 interacting proteins that interacted in at least three replicas and were previously unknown to interact. The number of unique proteins participating in these 48 interacting pairs was 47. The 48 interactions
FIGURE 3. Schematics of the procedure to choose the gold standard set in Challenge 2. A set of 47 proteins (the DREAM data set) was identified, that contained some protein—protein interaction pairs. Of the 1,128 possible protein interaction pairs among these 47 proteins, three nonoverlapping sets were identified. The “gold standard positive” set was composed of 48 protein—protein interactions not previously reported in the literature but validated in at least three out of nine independent Y2H experiments. The “gold standard negative” set was composed of 1,022 pairs not previously reported in the literature and never seen to interact in any of the nine replicates of the Y2H screen. The remaining 58 protein pairs forming the “Undecided set” were omitted from the positive and negative sets and ignored in the scoring because either the interaction was already published, it was not tested in our Y2H, it was detected as a positive in one or two replicates, or some combination of these situations. (Color is shown in online version.)

composed the gold standard positive set and formed the basis for the creation of the negative set as shown in Figure 3 and explained below. (The term positive set denotes the set of protein pairs that interact and negative set denotes the set of protein pairs that do not interact.) To arrive at the gold standard negative set, we enumerated the 1,128 possible protein interactions among 47 proteins. (In general, there are \(n(n+1)/2\) possible interaction among \(n\) proteins, counting self-interactions.) The gold standard negative set was composed of the subset of the 1,128 possible interactions for which (1) the interaction was negative in all replicates of the Y2H screen, and (2) there are no reports of that interaction in the literature. These criteria yielded a gold standard negative set consisting of 1,022 non-interacting protein pairs. The remaining 58 of the 1,128 possible protein pairs were omitted from the positive and negative sets and ignored in the scoring because either the interaction was already published, it was not tested in our Y2H, it was detected as a positive in one or two replicates, or some combination of these situations.

The protein–protein subnetwork challenge consisted of discriminating the protein pairs that interact (positives) from those that do not (negatives) among the list of 47 proteins. A guiding principle in the construction of the challenge was to highlight what may be called ab initio prediction, as opposed to literature mining, so interactions described in the literature were eliminated from the positive and negative sets, and predictions of published interactions were ignored in the scoring. All data in the
public domain was available to participants, but the unpublished Y2H screen on which the challenge was based was not disclosed for the duration of the challenge. Participants submitted a ranked list of the 1,128 protein pairs ordered from high to low confidence that a pair of proteins interacts. In other words, interacting pairs of proteins were to appear at the top of the list and non-interacting pairs were to appear at the bottom of the list. Only the 1,070 pairs that constituted the gold standard were evaluated.

The Five-Gene Network Challenges

In a fruitful experimental and computational collaboration, Dr. Maria Pia Cosma and Dr. Diego diBernardo, of the TIGEM Telethon Institute of Genetics and Medicine, in Naples, Italy, designed a synthetic five-gene network that was then integrated in the *Saccharomyces cerevisiae* genome. The synthetic-biology network, named IRMA (for In vivo Reverse engineering and Modeling Assessment) has since been used by its authors for “benchmarking” modeling and reverse-engineering methods. Data from this network, the design of which is schematically shown in Figure 4A, formed the basis of the DREAM2 five-gene network challenge. The network was designed in such a way that the promoter regions of the genes (represented by 1 pr1, 1 pr2, 4 pr, and 5 pr in Fig. 4A), was responsive to the network genes 1, 3, 4, and 5, as shown in Figure 4A. This was achieved by knocking out the endogenous genes that could potentially activate these promoter regions. A positive feedback loop (formed by 4 → 5 → 1 → 4), and two negative feedback loops (4 → 5 → 1 → 3 → 4 and 5 → 1 → 2 → 5) made for an interesting network dynamic.

When growing in a glucose medium, the organism harboring this synthetic circuit grows normally and reaches a steady state. When switched to a galactose reach medium, gene 2 is repressed, changing the network dynamics to a new steady state. In the change from glucose to galactose, the transient dynamics of the system can in principle be used to infer the topology of the network through reverse engineering algorithms. This was the rationale for two related DREAM2 challenges based on this synthetic network.

Two slightly different versions of the same network were built. In a cell cycle–dependent version of the network, gene 1, which encodes a transcription factor, was activated and translocated to the nucleus only in the late G1 phase of the cell cycle. In a second version of the network (a cell cycle–independent version), gene 1 was constitutively active independent of the cell cycle. Both versions of the network can be switched “on” and “off” by changing the concentration of galactose. In both cases the network is on when yeast cells are grown in galactose medium, and it is off when grown in glucose medium. DREAM2 participants attempted to infer the network structure from two independent datasets constituting two
independent subchallenges: *FiveGeneNet1* and *FiveGeneNet2*.

**FiveGeneNet1 Subchallenge**

The data set for this subchallenge was collected using the cell cycle–independent network and consisted of two time series based on quantitative PCR (qPCR) measurements of the five network genes. In each of the time series, yeast cultures were grown in a minimal medium with glucose for 10 hours and then shifted to a galactose medium. Cells were collected from the galactose medium at regular intervals, and the mRNA of the five genes in the synthetic network was quantified by qPCR. The culture was not cell-cycle synchronized in this experiment. The two time series provided in this subchallenge corresponded to samples taken at regular intervals of 20 minutes for 3 hr (time series qPCR_A) and for 5 hr (time series qPCR_B).

**FiveGeneNet2 Subchallenge**

In this subchallenge, the yeast populations containing the cell cycle–dependent network were cell-cycle synchronized by adding alpha factor to their medium and then exposed to two different treatments. In treatment 1, cells grew in glucose-supplemented minimal medium, whereas in treatment 2 cells grew in galactose-supplemented minimal medium. Following synchronization, the populations were released into the cell cycle, and samples were collected every 20 minutes for 3 hours. Whole genome expression profiles were obtained using Affymetrix chips. The datasets provided to the DREAM2 participants contained two time series (corresponding to the two different treatments) for 588 genes from the original Affymetrix microarray, which included the five genes in the synthetic network plus genes known in the literature to be regulated by the network genes. The identity of the 588 genes was kept anonymous, including which of the 588 were the five genes of interest.

For each of the two subchallenges, participants were given the freedom to submit predicted networks in one or more of a variety of formats in which the directionality of edges was either specified or not, and the signs of edges were either specified as excitatory or inhibitory. The resulting six categories were Undirected Unsigned, Undirected Excitatory, Undirected Inhibitory, Directed Unsigned, Directed Excitatory, and Directed Inhibitory. In all cases, participants submitted a list of network edges ordered by the confidence that the edge existed. The gold standards for the FiveGeneNet1 subchallenge are shown in Table 1 and are an example of the format requested from the participants. (The gene numbers correspond to the graph of Fig. 4B). In the undirected submissions, the order of the first and second columns was inessential, but if the pair gene_\text{k} gene_\text{l} appeared in the list, then the pair gene_\text{l} gene_\text{k} should not have appeared. In the directed submissions, the first and second columns indicated the source and destination of a directed edge, respectively. In both directed and undirected submissions, the third column indicated an arbitrary measure of confidence of the existence of an edge. The third column was requested at submission to verify that the order of the submitted pairs was indeed in the order of higher to lower confidence. In other words, regulatory interactions were to appear at the top of the list and non-interactions were to appear at the bottom of the list.

In the FiveGeneNet2 subchallenge we used the same gold standards as the FiveGeneNet1 subchallenge described above. Of course, the underlying network is the same; only the measurements had changed. We disregarded all predicted edges for which both genes were not members of the five-gene synthetic network.

**The in Silico Network Challenges**

*In silico* networks formed the basis of a set of challenges in which participants attempted to infer the underlying network structure from simulated data. These challenges, while somewhat contrived, enabled the assessment of network reconstructions for which the underlying
### TABLE 1. Gold standards for the six categories in the five-gene network challenge

<table>
<thead>
<tr>
<th>Undirected_Unsigned</th>
<th>Undirected_Signed-Excitatory</th>
<th>Undirected_Signed-Inhibitory</th>
</tr>
</thead>
<tbody>
<tr>
<td>gene_1 gene_2 1</td>
<td>gene_1 gene_2 1</td>
<td>gene_2 gene_5 1</td>
</tr>
<tr>
<td>gene_1 gene_3 1</td>
<td>gene_1 gene_3 1</td>
<td>gene_3 gene_4 1</td>
</tr>
<tr>
<td>gene_1 gene_4 1</td>
<td>gene_1 gene_4 1</td>
<td>gene_1 gene_1 0</td>
</tr>
<tr>
<td>gene_1 gene_5 1</td>
<td>gene_1 gene_5 1</td>
<td>gene_1 gene_2 0</td>
</tr>
<tr>
<td>gene_2 gene_5 1</td>
<td>gene_4 gene_5 1</td>
<td>gene_1 gene_3 0</td>
</tr>
<tr>
<td>gene_3 gene_4 1</td>
<td>gene_1 gene_1 0</td>
<td>gene_1 gene_4 0</td>
</tr>
<tr>
<td>gene_4 gene_5 1</td>
<td>gene_5 gene_1 0</td>
<td>gene_1 gene_5 0</td>
</tr>
<tr>
<td>gene_1 gene_1 0</td>
<td>gene_2 gene_3 0</td>
<td>gene_2 gene_2 0</td>
</tr>
<tr>
<td>gene_2 gene_2 0</td>
<td>gene_2 gene_4 0</td>
<td>gene_2 gene_3 0</td>
</tr>
<tr>
<td>gene_2 gene_3 0</td>
<td>gene_2 gene_5 0</td>
<td>gene_3 gene_1 0</td>
</tr>
<tr>
<td>gene_2 gene_4 0</td>
<td>gene_3 gene_2 0</td>
<td>gene_3 gene_3 0</td>
</tr>
<tr>
<td>gene_3 gene_5 0</td>
<td>gene_3 gene_3 0</td>
<td>gene_3 gene_5 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Directed_Unsigned</th>
<th>Directed_Signed-Excitatory</th>
<th>Directed_Signed-Inhibitory</th>
</tr>
</thead>
<tbody>
<tr>
<td>gene_1 gene_2 1</td>
<td>gene_1 gene_2 1</td>
<td>gene_2 gene_5 1</td>
</tr>
<tr>
<td>gene_1 gene_3 1</td>
<td>gene_1 gene_3 1</td>
<td>gene_3 gene_4 1</td>
</tr>
<tr>
<td>gene_1 gene_4 1</td>
<td>gene_1 gene_4 1</td>
<td>gene_1 gene_1 0</td>
</tr>
<tr>
<td>gene_2 gene_5 1</td>
<td>gene_4 gene_5 1</td>
<td>gene_1 gene_3 0</td>
</tr>
<tr>
<td>gene_3 gene_4 1</td>
<td>gene_1 gene_1 0</td>
<td>gene_1 gene_4 0</td>
</tr>
<tr>
<td>gene_4 gene_5 1</td>
<td>gene_5 gene_1 0</td>
<td>gene_1 gene_5 0</td>
</tr>
<tr>
<td>gene_1 gene_1 0</td>
<td>gene_2 gene_3 0</td>
<td>gene_2 gene_2 0</td>
</tr>
<tr>
<td>gene_1 gene_5 0</td>
<td>gene_2 gene_2 0</td>
<td>gene_2 gene_2 0</td>
</tr>
<tr>
<td>gene_2 gene_1 0</td>
<td>gene_2 gene_3 0</td>
<td>gene_3 gene_1 0</td>
</tr>
<tr>
<td>gene_2 gene_2 0</td>
<td>gene_2 gene_4 0</td>
<td>gene_2 gene_4 0</td>
</tr>
<tr>
<td>gene_2 gene_3 0</td>
<td>gene_2 gene_5 0</td>
<td>gene_3 gene_1 0</td>
</tr>
<tr>
<td>gene_2 gene_4 0</td>
<td>gene_3 gene_2 0</td>
<td>gene_3 gene_3 0</td>
</tr>
<tr>
<td>gene_3 gene_5 0</td>
<td>gene_3 gene_3 0</td>
<td>gene_3 gene_5 0</td>
</tr>
<tr>
<td>gene_4 gene_1 0</td>
<td>gene_4 gene_1 0</td>
<td>gene_4 gene_3 0</td>
</tr>
<tr>
<td>gene_4 gene_2 0</td>
<td>gene_4 gene_2 0</td>
<td>gene_4 gene_4 0</td>
</tr>
<tr>
<td>gene_4 gene_3 0</td>
<td>gene_4 gene_3 0</td>
<td>gene_4 gene_5 0</td>
</tr>
<tr>
<td>gene_4 gene_4 0</td>
<td>gene_4 gene_4 0</td>
<td>gene_5 gene_1 0</td>
</tr>
<tr>
<td>gene_5 gene_2 0</td>
<td>gene_5 gene_2 0</td>
<td>gene_5 gene_2 0</td>
</tr>
<tr>
<td>gene_5 gene_3 0</td>
<td>gene_5 gene_3 0</td>
<td>gene_5 gene_3 0</td>
</tr>
<tr>
<td>gene_5 gene_4 0</td>
<td>gene_5 gene_4 0</td>
<td>gene_5 gene_4 0</td>
</tr>
<tr>
<td>gene_5 gene_5 0</td>
<td>gene_5 gene_5 0</td>
<td>gene_5 gene_5 0</td>
</tr>
</tbody>
</table>

*The table exemplifies the submission format. In the Undirected submissions, the order of the first and second columns was inessential. In the Directed submissions, the first and second columns represented, respectively, the source and destination of a directed edge. The third column, indicating an arbitrary measure of confidence of the corresponding edge that was in the actual network, was requested at submission to verify that the order of the submitted pairs was indeed in the order of higher to lower confidence.*
network, dynamical equations, and noise model were known with certainty to the evaluator. In this way, we trade realism of the problem for accuracy of the gold standard.

Three *in silico* networks were created and endowed with deterministic dynamics without noise using the methods described by Mendes and collaborators.\textsuperscript{16} Dr. Pedro Mendes provided the networks and the corresponding data. The phenomenological models of gene regulatory networks had Hill-type kinetics for induction/repression, and only additive interactions were assigned. This challenge consisted of three subchallenges.

**InSilico1 Subchallenge**

The challenge consisted of reverse engineering a network of 50 nodes and 98 directed edges. The network was constructed to have Erdos-Renyi topology (see Fig. 5A).

**InSilico2 Subchallenge**

The challenge consisted of identifying a network of 50 nodes and 99 directed edges. The network was constructed to have scale-free topology (see Fig. 5B).

**InSilico3 Subchallenge**

The challenge consisted of reverse engineering an integrated gene-protein-metabolite network consisting of 24 metabolites, 23 proteins, and 20 genes connected through 146 directed edges. The network was constructed manually to mimic known biochemical network structures (see Fig. 5C).

In the first two networks, an edge represented a causal relationship between the expression levels of two genes. For the latter network, edges represented processes such as protein translation, protein–protein interactions, and enzyme-mediated metabolic reactions.

Simulated data for the *InSilico1* and *InSilico2* subchallenges mimicked typical functional genomic experiments, such as gene expression microarray data, or quantitative PCR data, with the additional advantage that the number of *in silico* experiments to the number of genes was larger than usual. The data for the reconstruction of the *InSilico3* network also contained the equivalent of the protein and metabolite concentrations, in addition to the mRNA abundance. For the first two subchallenges, the data provided to the participants consisted of the following.

- **Knock-down data**: This dataset contained steady-state mRNA abundance for the 50 genes under 51 experimental conditions. In addition to the expression of the “wild-type” strain, steady-state expression profiles of the 50 heterozygous (+/–) knock-down strains were provided.
- **Knock-out data**: This dataset contained steady-state mRNA abundance for the 50 genes under 51 experimental conditions. In addition to the expression of the wild-type strain, steady-state expression profiles of the 50 heterozygous (–/–) null-mutant strains were provided.
- **Time-series data**: This dataset contained the mRNA concentration for the 50 genes as their abundance evolved dynamically in the wild-type strain (+/+). Time courses (trajectories) of the network recovering from several external perturbations induced by different initial conditions were provided. There were 23 time courses differing in the initial conditions; each time course was measured at 26 time points.

For the third subchallenge, the data was slightly different. It consisted of the following.

- **Knock-down data**: This dataset contained steady-state levels of metabolites, proteins, and mRNA for the wild-type and the 20 heterozygous knock-down strains for each gene (+/–).
- **Knock-out data**: This dataset contained steady-state levels of metabolites, proteins, and mRNA for the wild-type and the 20 null-mutant strains for each gene (–/–). The knock-out of the gene encoding G14
was in silico “lethal” (i.e., most of the steady states turned out to be zero).

- Time-series data: This dataset contained time courses (trajectories) of the network recovering from several external perturbations. There were 23 time series differing in the initial conditions; each time series had 26 time points.
Participants were given the freedom to submit predicted networks in one or more of a variety of formats in which the directionality of edges was either specified or not, and the signs of edges were either specified as excitatory/inhibitory, or not. As was the case in the five-gene network challenge, these options resulted in the following six categories per subchallenge: Undirected Unsigned, Directed Unsigned, Undirected Excitatory, Undirected Inhibitory, Directed Excitatory, Directed Inhibitory. In all cases, participants submitted a ranked list of network edges ordered from high to low confidence that an edge was present. In other words, regulatory interactions were to appear at the top of the list and non-interactions were to appear at the bottom of the list.

The convention for the format of the submissions was not straightforward for the InSilico3 subchallenge due to the existence of enzyme-mediated reactions and bi-molecular reactions. The specific conventions used to format the prediction files can be found in the DREAM2 in silico challenges data description page of the DREAM website.17

The Genome Scale Network Challenge

A long-standing question in the reverse engineering community is: To what extent can a gene regulatory network be reverse engineered from steady-state gene expression profiles? A compendium of more than 500 normalized E. coli Affymetrix microarrays representing about 200 distinct conditions18 formed the basis of a challenge in which participants attempted to reverse engineer the E. coli gene regulatory network from microarray data. Of the more than 500 original microarrays, more than 300 were collected at the laboratories of James J. Collins and Tim Gardner at Boston University, and more than 200 were downloaded from the Gene Expression Omnibus (GEO) website. The approximately 500 experimental conditions were concentrated mainly on antibiotic, SOS, and stress response. The selected 300 microarrays reasonably covered many stress pathways. The currently known E. coli gene regulatory network as reported by RegulonDB,19 arguably the most complete gene regulatory network database for any organism, was the source of the gold standard set for this challenge. The participants were not informed that the data pertained to E. coli. Genes in the microarray data were anonymized by removing the labels and adding a small amount of Gaussian noise so that the data sources could not be easily identified by Internet searches on the expression values. Included in the microarray data were 3,456 coding regions that included all genes in RegulonDB and randomly chosen additional genes. Among the included genes were 320 known and putative transcription factors annotated in RegulonDB.

To arrive at the gold standard, we began with the then current annotated transcription factor-gene regulation as curated in RegulonDB version 4(beta). To this we added publicly available sigma factor regulation available from the RegulonDB website, and removed self-regulation. The gold standard network so constructed contained 1,095 genes and 152 transcription factors (some of them counted among the 1,095 genes) with annotated interactions, and 3,102 edges, about one third of which were sigma factor interactions. Some caveats related to using RegulonDB as a gold standard for assessment of predicted networks are illustrated in Figure 6. At best, the RegulonDB network is incomplete. Based on ChIP-qPCR data of three transcription factors, it was estimated that RegulonDB is 85% complete.20 Therefore, some false positives may be incorrectly assigned. The functional depth of the microarray compendium may not be sufficient to recover large portions of the RegulonDB network. It is easy to imagine the situation in which a high-quality network prediction can have a low recall but high precision.

Participants were given the freedom to submit predicted directed networks in which the signs of edges were either specified as excitatory/inhibitory or not specified. These options resulted in the following categories: Directed
FIGURE 6. Relationships among predicted, gold standard, and actual networks in Challenge 5. The “gold standard” set consisted of the transcription factor-target interactions as curated in RegulonDB v.4(beta) enhanced with publicly available sigma factor regulation also available from the RegulonDB website and with self-regulation removed. This gold standard is a subset of the actual E. coli transcriptional network, represented the “Real network” in the figure. The “Predicted network” set may contain some incorrectly assigned false positives. Insufficiently sampled conditions in the provided data may have lead to predictions with a low recall but high precision. (Color is shown in online version.)

Unsigned, Directed Excitatory, and Directed Inhibitory. The direction was specified by the fact that for the resulting network, any edge had to originate from a TF. In all cases, participants submitted a ranked list of network edges ordered from high to low confidence that an edge is present. In other words, regulatory interactions were to appear at the top of the list and non-interactions were to appear at the bottom of the list.

Participants predicted a network of 3,456 genes × 320 transcription factors. For scoring, this network was mapped to the 1,095 genes × x 152 transcription factors in the gold standard. Transcription factor-gene edges that are present in RegulonDB form the gold standard positive set. The gold standard negative set was defined as the transcription factor-gene entries that were absent in RegulonDB. It is clear, however, that absence of an edge in RegulonDB doesn’t mean that edge is inexisten in the actual network.

Assessment of Predicted Networks

In order to assess how well a predicted network approximates the gold standard network that it attempts to recapitulate we require a metric, or a suite of metrics. Broadly speaking, an assessment of a predicted network should express the concept of distance between the predicted network and the gold standard network. Consider, for example, the edit distance (or Hamming distance), which can be defined on a network as the number of corrections (e.g., added or removed edges) to transform one network into another. Edit distance is an attractive metric given its simplicity; however, it does not discriminate between the two types of errors that occur when predicting a network edge, false positives (FP) and false negatives (FN). Since biological networks are sparse, containing many fewer edges than is theoretically possible, it is important to characterize a predicted network with respect to each type of error. Since network prediction is similar to other types of binary prediction problems, it is advantageous to employ a generalized assessment of prediction accuracy.

As posed to the DREAM2 participants, network prediction can be thought of as a binary classification task in which every potential network edge is classified as either present or absent. In some cases we could have had three-class predictions. For example, in a directed regulatory network an edge could be either (1) absent, (2) present and excitatory, or (3) present and inhibitory. The assessment of prediction of networks whose edges could belong to more than two classes, however, posed some nontrivial technical difficulties. Therefore we decided to pose the network challenges as binary problems. The simplification of problems to binary classes came at the price of complicating the categories of submission. In the example above, rather than the category Directed Signed, for which the predicted edges could belong to more than two classes, we created two categories, Directed Excitatory and Directed Inhibitory.
The advantage of using binary classes is that binary classification is a standard paradigm in the field of machine learning with well-established evaluation metrics. Because there are two type of errors (also, two types of correct predictions), it is common to simultaneously explore complementary metrics over a range of parameter values (e.g., varying the cutoff of the decision boundary). One such pair of complementary metrics is precision and recall. Precision is a measure of fidelity, whereas recall is a measure of completeness. Another common pair of metrics is true positive rate (TPR) and false positive rate (FPR), the axes of the receiver operator characteristic (ROC) curve. These concepts are reviewed below.

To Threshold or Not to Threshold?

Any algorithm designed to decide the class that each element in a set belongs to (say Positive and Negative elements) will have to output two sets: the set of elements predicted to be Positive, and the set of elements predicted to be Negative. At some point the researcher is confronted by the need to choose a cutoff that determines the boundary between classes. This cutoff is typically controlled by one or more parameters intrinsic to the algorithm. The choice of this cutoff is usually arbitrary, such as, for example, determining a $P$-value below which a prediction is deemed significant.

One of the problems in assessing methods that use a threshold is that the evaluation will simultaneously assess the choice of the threshold along with the method used for the predictions. If a method is excellent but the choice of the cutoff is suboptimal, the evaluation might give a bad score to the method. If instead we free the evaluation from a choice of the cutoff, the assessment will only consider the merits of the classification algorithm. The latter is the strategy that we chose for the assessment of DREAM2 predictions. Of course, availability of an objective approach to determine an optimal metric is an important aspect of reverse engineering. For instance, an algorithm may be far superior to others in the overall thresholdless metric. Yet, in the absence of a reasonable approach to determine the threshold (which may be context-dependent), the algorithm may behave poorly in practical conditions. This potential oversight will be considered in future challenge editions.

Format of Predictions

If the network predictions included the choice of a threshold, the submissions would have resulted in two sets, one class with pairs of nodes deemed to have an edge, and the other set with the pair of nodes predicted not to have an edge. Instead, DREAM2 participants were asked to submit their predictions in the form of an ordered list of predicted network edges. The list was constructed in decreasing order of reliability that an edge is in the network. In this way, the first list entry is the network edge predicted with highest confidence, and the last list entry is the network edge predicted with the lowest confidence that the edge belongs in the network. Let $k$ be a pointer to the current list entry, $k = 1, \ldots, L$, where $L$ is the length of the list. If a particular $k$ was chosen as a threshold to determine the boundary of positive and negative sets, then the edges from 1 to $k$ would be the set of network edges predicted at cutoff $k$. Rather than considering a particular cutoff $k$, we considered all possible choices of $k$. In other words, $k$ is a parameter that incorporates progressively more edges in the predicted network as $k$ is increased from 1 to $L$. Suppose that the challenge demanded the prediction of $T$ total edges (for example, for a network with $n$ nodes, a challenge requesting the prediction of a directed network without self-interactions would have $T = n(n - 1)$ edges). If $L = T$, then $k$ truly ranges from a positive set with just one edge ($k = 1$), to a positive set including all edges ($k = T$). However, the participant may submit a truncated list ($L < T$). In such case, the edges not included in the submitted prediction should be “added” in random order at the end of the list to complete the submission. We say
“added” in quotations because the addition of missing edges takes place in an analytical way. In other words, the final assessment was made on the average performance that would result if the missing edges were added in all possible orders (see Appendix).

**Precision-Recall Curve**

As explained above, for each value of the parameter $k$ indexing the position of an entry in the submitted edge list, we have a concrete network prediction. On this prediction a measure of fidelity of the positive set can be defined as the *precision* at depth $k$ of the prediction as

$$\text{prec}(k) = \frac{TP(k)}{TP(k) + FP(k)} = \frac{TP(k)}{k},$$

where $TP(k)$ is the number of true positives (edges predicted to be in the gold standard network that actually are in it), and $FP(k)$ is the number of false positives (edges predicted to be in the gold standard network that actually are not in it), both at depth $k$ in the list (note that $TP(k) + FP(k) = k$). Another useful measure of accuracy of a prediction is the recall, a measure of completeness of a prediction, defined as

$$\text{rec}(k) = \frac{TP(k)}{P},$$

where $P$ is the number of actual positives (i.e., the number of actual edges in the network), and therefore is independent of $k$. The values of recall and precision range from zero to one, with one being the theoretically optimal value for precision, and $\min(k/P, 1)$ being the theoretical optimum for recall at depth $k$. As $k$ increases from 1 and $T$, the precision-recall curve graphically explores changes in accuracy as we add more edges to our putative positive set.

**ROC Curve**

The receiver operating characteristic (ROC) curve graphically explores the tradeoff between the true positive rate $TPR(k)$ and the false positive rate $FPR(k)$, as $k$ ranges from 1 and $T$. The true positive rate at depth $k$ is computed as

$$TPR(k) = \frac{TP(k)}{P},$$

and represents the fraction of positives that are correctly predicted at depth $k$ in the submitted list. Note that $TPR(k)$ is equivalent to $rec(k)$. The false positive rate at depth $k$ in the submitted list, $FPR(k)$, is defined as

$$FPR(k) = \frac{FP(k)}{N},$$

where $N$ is the total number of node pairs that are not connected by edges, and represents the fraction of negatives that are incorrectly predicted as positives at depth $k$. Both $TPR(k)$ and $FPR(k)$ range from zero to one. As $k$ increases from 1 to $T$, there is a tradeoff between $FPR(k)$ and $TPR(k)$. The perfect true positive rate will be achieved at $k = T$, by predicting the completely connected network (indeed, $TP(k = T) = P$, and $TPR(k = T) = 1$). However, $FPR(k = T)$ would be the highest (equal to 1) due to the fact that at $k = T$ all the edges that are actually negatives would be predicted positive ($FP(k = T) = N$). Thus, the tradeoff between $TPR(k)$ (which we desire to be high) and $FPR(k)$ (which we desire to be low) can be inspected directly from the ROC curve.

**Area Under the Curves**

To compare the submitted predictions against a corresponding gold-standard, we use a method based on the “Area Under the Curve,” a rather standard measure of classification performance in the machine learning community. The area under the ROC curve (AUROC) is a single number that summarizes the tradeoff between $TPR(k)$ and $FPR(k)$ as $k$ is varied. To compute it, we simply plot the points $(TPR(k), FPR(k))$ in a $TPR$ versus $FPR$ plane and compute the integral of the curve that results from a linear interpolation between the points. A perfect network prediction would yield an AUROC of one.

The area under the precision-recall curve (AUPR) is a single number that summarizes
the precision-recall tradeoff. To compute it, we simply plot the points \((\text{prec}(k), \text{rec}(k))\) in a precision versus recall plane and compute the integral of the curve that results from a nonlinear interpolation between the points. The rational for the nonlinear form of the interpolation, and an algorithm to compute the AUPR used in DREAM2, is presented in the Appendix. A perfect network prediction would yield an AUPR of one.

The value of the area under the curves depends on the particular trajectory taken by the points in the precision-recall curve and the ROC curve. It could very well happen that two predictions have the same AUPR and vastly different AUROC. This is clearly seen in
Figure 7A, where we show the results of creating a large number of random network predictions for the five-gene-network challenge, in the category Undirected Unsigned. It is clear that for a given value of AUROC, there is a range of possible values of the AUPR. For example, the ensemble of predictions with AUROC = 0.7, can have values of the AUPR between 0.55 and 0.8. Therefore both AUPR and AUROC can be used as measures of the performance of a network submission, as each gives a subtly different summarization of the TP(k) and FP(k) measures over the whole range of k.

Scoring a Network Submission

To gain a sense of the statistical significance of a particular area under the curve, we require a null hypothesis from which a P-value could be computed. A reasonable null hypothesis is a random list of network edge predictions. For many such random predictions, we computed the AUROC and the AUPR. By sampling the space of random predictions (see Fig. 7A), the joint probability density function of obtaining simultaneously a pair (AUROC, AUPR) could, in principle, be estimated. The P-value for an algorithm that simultaneously yields areas under the curve aROC and aPR could be computed as the probability of AUROC > aROC and AUPR > aPR. (Other definitions of a P-value for a bivariate distribution are possible.) In practice, sampling this bivariate probability density with sufficient precision was unfeasible because the space of high AUROC and high AUPR was extremely unlikely.

Instead we took the strategy of computing independently the marginal P-values:

\[ p_{ROC}(a_{ROC}) = \text{Prob}(\text{AUROC} > a_{ROC}) \]

and

\[ p_{PR}(a_{PR}) = \text{Prob}(\text{AUPR} > a_{PR}) \]

and we defined the score of a submission with areas under the curve aROC and aPR as

\[ \text{score}(a_{PR}, a_{ROC}) = -\log_{10}[p_{PR}(a_{PR})p_{ROC}(a_{ROC})]. \]

With this definition, the higher the score, the better the merits of the submission.

We finish this section with a technical but important point. In many of the challenges the submitted networks obtained values of the AUROC and AUPR that were too big to be reached by sampling the space of random predictions. We addressed this problem with an empirical approach that is illustrated in Figures 7B and 7C. Shown in blue in Figure 7B is the probability density function of the AUPR that resulted from sampling 100,000 random predictions for the InSilico1 network challenge, category Undirected Unsigned. The maximum value of AUPR sampled was 0.2, but the AUPR obtained by the best performer team was 0.6.

Estimation of a P-value for this team entails integrating the distribution between 0.6 and 1, but we didn’t have any sampled value of the probability distribution in this range. (It should not come as a surprise that some teams performed much more accurately than random chance.) Therefore we parameterized the null probability with a function that approximates it very well, as shown in the black line that fits the blue points extremely well. The approximation was given by stretched exponentials with different parameters to the right and left of the mode of the distribution, with functional form

\[ \text{pdf}(x) = \begin{cases} h_{\text{max}} \exp\left[ -b_\geq(x - x_{\text{max}})^{c_\geq} \right] & \text{for } x \geq x_{\text{max}} \\ h_{\text{max}} \exp\left[ -b_\leq(x_{\text{max}} - x)^{c_\leq} \right] & \text{for } x < x_{\text{max}} \end{cases} \]

For Figure 7B, the parameters were given by \( h_{\text{max}} = 49.9655, x_{\text{max}} = 0.0785, c_\geq = 2.8031, b_\geq = 470,450, c_\leq = 1.1565, \) and \( b_\leq = 198.86. \) With this functional form, the resulted P-value could be computed to yield \( \sim 10^{-41}. \) A similar procedure done for the distribution of AUROC (Fig. 7C) yielded a P-value of \( \sim 10^{-41}. \) In this way, the score for this team was 61, an extremely satisfactory performance.

We applied the procedure just described to every category in all challenges. In this way we could score the performance of all the DREAM2 submissions.
FIGURE 8. The statistics of submissions to the five DREAM2 challenges. The most popular of the challenges was the In Silico challenge, followed by the BCL6 target prediction challenge. (Color is shown in online version.)

Best Performers

Best performers were selected in each category of each challenge. They were identified on the basis of the following criteria:

(i) AUPR P-value ≤ 0.05.
(ii) AUROC P-value ≤ 0.05.
(iii) Highest score of its category among the teams for which conditions (i) and (ii) are satisfied.

Considered as a sample of the community performance in reverse engineering problems, the compilation of DREAM2 submissions constitutes a true community-wide experiment that can be studied as data in its own right. In the following sections we report on the assessment of the performance of the participants in the different DREAM2 challenges.

Results

Statistics of Submissions

On July 22, 2007 the challenges were posted on the DREAM website, and submissions in response to the challenges were accepted until October 23, 2007. During these three months, 109 teams, from 17 different countries (Argentina, Belgium, Canada, China, France, Germany, Hungary, India, Italy, Japan, Korea, Russia, Singapore, Sweden, Switzerland, UK, USA) downloaded the challenges. Of those, 36 teams submitted for evaluation a total of 110 predictions. The statistics of submissions can be seen in Figure 8. The most popular of the challenges was the In Silico challenge, followed by the BCL6 target prediction challenge. In the following subsections we will present analyses of performance for a subset of all these submissions. The remaining results can be obtained from the DREAM website: http://wiki.c2b2.columbia.edu/dream/.

The BCL6 Transcriptional-Target Challenge

Eleven teams submitted predictions for the BCL6 target discovery challenge. Predictions were assessed as described in the section Assessment of Predicted Networks and shown in Table 2. We can see that the best performing team according to the criteria delineated in the previous section was Team 2. Note, however,
### TABLE 2. Performance of submissions to the BCL6 target identification challenge

<table>
<thead>
<tr>
<th>Team</th>
<th>1st</th>
<th>2nd</th>
<th>5th</th>
<th>20th</th>
<th>AUPR</th>
<th>P-value</th>
<th>AUROC</th>
<th>P-value</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>1</td>
<td>1</td>
<td>0.83</td>
<td>0.83</td>
<td>0.69</td>
<td>$1.2 \times 10^{-11}$</td>
<td>0.852</td>
<td>$3.3 \times 10^{-14}$</td>
<td>24.38</td>
</tr>
<tr>
<td>Team 2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.80</td>
<td>0.72</td>
<td>$1.4 \times 10^{-12}$</td>
<td>0.847</td>
<td>$7.4 \times 10^{-14}$</td>
<td>24.99</td>
</tr>
<tr>
<td>Team 3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.67</td>
<td>0.67</td>
<td>$4.9 \times 10^{-11}$</td>
<td>0.83</td>
<td>$5.4 \times 10^{-13}$</td>
<td>22.57</td>
</tr>
<tr>
<td>Team 4</td>
<td>0.33</td>
<td>0.50</td>
<td>0.38</td>
<td>0.45</td>
<td>0.40</td>
<td>0.002</td>
<td>0.72</td>
<td>$1.3 \times 10^{-6}$</td>
<td>8.53</td>
</tr>
<tr>
<td>Team 5</td>
<td>1</td>
<td>0.67</td>
<td>0.62</td>
<td>0.41</td>
<td>0.38</td>
<td>0.005</td>
<td>0.63</td>
<td>0.0022</td>
<td>4.97</td>
</tr>
<tr>
<td>Team 6</td>
<td>1</td>
<td>1</td>
<td>0.45</td>
<td>0.30</td>
<td>0.33</td>
<td>0.057</td>
<td>0.54</td>
<td>0.21</td>
<td>1.93</td>
</tr>
<tr>
<td>Team 7</td>
<td>0.50</td>
<td>0.67</td>
<td>0.45</td>
<td>0.27</td>
<td>0.31</td>
<td>0.11</td>
<td>0.52</td>
<td>0.30</td>
<td>1.49</td>
</tr>
<tr>
<td>Team 8</td>
<td>0.09</td>
<td>0.13</td>
<td>0.21</td>
<td>0.26</td>
<td>0.25</td>
<td>0.73</td>
<td>0.50</td>
<td>0.51</td>
<td>0.43</td>
</tr>
<tr>
<td>Team 9</td>
<td>1</td>
<td>0.40</td>
<td>0.55</td>
<td>0.24</td>
<td>0.29</td>
<td>0.23</td>
<td>0.46</td>
<td>0.81</td>
<td>0.73</td>
</tr>
<tr>
<td>Team 10</td>
<td>0.11</td>
<td>0.07</td>
<td>0.10</td>
<td>0.21</td>
<td>0.20</td>
<td>0.998</td>
<td>0.39</td>
<td>0.99</td>
<td>0.005</td>
</tr>
<tr>
<td>Team 11</td>
<td>0.04</td>
<td>0.06</td>
<td>0.09</td>
<td>0.16</td>
<td>0.18</td>
<td>1</td>
<td>0.28</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

*The precision of the 1st, 2nd, 5th and 20th correct predictions for each team are listed in the 2nd to 5th columns. AUPR and AUROC indicate, respectively, the area under the Precision-Recall curve and the area under the ROC curve. The last column lists the score, defined as minus the logarithm, to the base 10, of the product of the AUPR P-value (7th column) and the AUROC P-value (9th column).*

that the AUROC of the second best scoring team (Team 1) was larger than that of Team 2, and that the difference in score between Team 2 and Team 1 is rather small. In view of these observations, the uncertainties of our $P$-value calculations, and the non-uniqueness of the evaluation methodology, we decided that the performances of Team 1 and 2 were, by all intents and purposes, a tie. It is also interesting to observe that the jump of 14 units of score between the 3rd and 4th better scoring teams (Team 3 and Team 4) was much bigger than the jump of 1.8 units between the 2nd and 3rd better scoring teams (Team 1 and Team 3).

To visually compare the performance of the submissions, we performed agglomerative unsupervised hierarchical clustering on the vectors $prec(k)$ for $k = 1, \ldots, 200$ (see Fig. 9), the precision at depth $k$ in the submitted list. (See the legend to Fig. 9 for further explanation.) The precision of the three best performers nicely clustered with the precision of the gold standard (a perfect prediction should have a $prec(k)$ identical to the gold standard), whereas the remaining eight teams agglomerated in a separate cluster, with Teams 6 to 11 performing not significantly better than random.

The AUPR gives a sense of the typical precision of a method if we don’t have a particular threshold or cutoff for our predictions. In real applications, however, we do need to use some criteria to separate the two classes under scrutiny. For this challenge, we need to choose a $k_c$, and call all the predictions from 1 to $k_c$ as targets, and call the predictions at depth $k > k_c$ as non-targets. Columns 2 through 5 in Table 2 give us information on the precision at a given depth in the list. The entries in these columns indicate the precision at the n-th correct prediction. As the precision is computed as $prec(k) = n/k$, then the depth $k$ to achieve the n-th correct prediction is $k = n/prec(k)$. In order to get the 20th BCL6 target (representing a recall of 38%), we need to go to a depth of 24, 25, 30, 44, 49, 67, 74, 77, 83, 95, and 125, for teams 1 through 11. The average precision expected for a random prediction is 27%. Table 2 shows that at recall 38%, the precision of Teams 1 through 3 is quite better than what is expected by chance, while the precision of Teams 6 through 11 is pretty much what is expected by chance. Teams 4 and 5 are marginally better than chance.

The overall picture is that Teams 1, 2, and 3 are clearly the best performers in this challenge.
FIGURE 9. Heat map and hierarchical cluster analysis of the precision for the gold standard and the 11 submissions for the BCL6 target identification challenge. Precision is encoded in a color scale (shown in the palette at the upper right of the figure) going from blue (representing the precision for a random submission), to red (representing a submission with a precision of 1). The precision is parameterized by the prediction number (i.e., the depth on the prediction list) at which the cutoff is made. The gold standard has prediction 1 (saturated red) up to the position 53 in the list, below which the decoys started to degrade the prediction, down to the prediction number 200, at which the precision was 53/200 (saturated blue), the average precision expected at any cutoff in random predictions. (Color is shown in online version.)

(We included Team 3 even though it does not strictly meet the requirement for best performer given earlier. However, it is clear from the present analysis that Team 3 belongs in the group of best performers.)

The best performing teams were (team leaders are underlined):

- Team 2: Vinsensius B. Vega, Xing Yi Woo, Habib Hamidi, Hock Chuan Yeo, Zhen Xuan Yeo, Guillaume Bourque, and Neil D. Clarke, from the Genome Institute of Singapore
- Team 3: Matti Nykter, Harri Lähdesmäki, Alistair Rust, Vésteinn Thorsson, and Ilya Shmulevich, from The Institute for Systems Biology.

These teams discuss their best performing strategies in their respective chapters later in this volume.21–23 Even though the methods used by these three teams were different in many ways, it is interesting to highlight important similarities in their methods.

1. Avoid giving too much weight to “known” binding sites. Team 1 chose not to use the “known” BCL6 binding site. Team 2 used a weighted consensus of attributes based on gene expression experiments, sequence motifs, and Gene Ontology. Team 3 used a weighted consensus of attributes based on gene expression experiments, sequence motifs, and Gene Ontology. Therefore possible artifacts that the “known” BCL6 motifs could have incorporated in the analyses may have been compensated by the other information included in the consensus.

2. Use gene expression that is only relevant to the transcription factor under scrutiny. Of the 336 gene expression experiments Teams 1, 2, and 3 only used the gene expression data
TABLE 3. Performance of submissions to the protein–protein interaction subnetwork challenge

<table>
<thead>
<tr>
<th>Team</th>
<th>Precision at nth correct prediction</th>
<th>AUROC</th>
<th>AUROC</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>0.06 0.09 0.10 0.10</td>
<td>0.08 0.01 0.63</td>
<td>$1.3 \times 10^{-3}$</td>
<td>4.9</td>
</tr>
<tr>
<td>Team 2</td>
<td>0.14 0.10 0.07 0.06</td>
<td>0.05 0.31 0.56</td>
<td>0.077</td>
<td>1.6</td>
</tr>
<tr>
<td>Team 3</td>
<td>0.33 0.02 0.04 0.05</td>
<td>0.05 0.31 0.49</td>
<td>0.59</td>
<td>0.7</td>
</tr>
<tr>
<td>Team 4</td>
<td>0.50 0.25 0.03 0.04</td>
<td>0.05 0.31 0.48</td>
<td>0.68</td>
<td>0.7</td>
</tr>
<tr>
<td>Team 5</td>
<td>0.14 0.15 0.04 0.04</td>
<td>0.04 0.84 0.48</td>
<td>0.68</td>
<td>0.2</td>
</tr>
</tbody>
</table>

aThe columns are as in Table 2.

collected under conditions relevant to the BCL6 biology and neglected the rest of the data.

(3) Use previously known target genes as prior information. All three teams collected a training dataset of previously known BCL6 targets and used it to train a machine learning algorithm, which was later used to classify the test gene set.

We asked the remaining teams to voluntarily share with us what their methodology was, so that we could also extract lessons from the strategies that performed sub-optimally. Only Team 6 responded. In their approach they first looked for matches to a BCL6 weight matrix in the promoter regions of the 200 candidate genes. Those genes with a motif match above a relatively lenient threshold were then ranked based on how well their mRNA expression profile across all phenotypes correlated with the average expression profile of the genes. In their methodology, this team seems to have taken the opposite approach to the items shared by the best performing teams and enumerated above. Neither their AUPR nor their AUROC yielded values significantly different from random predictions.

The Protein-Protein Subnetwork Challenge

Five teams submitted predictions for the protein–protein interaction subnetwork challenge. Predictions were assessed as described in the section Assessment of Predicted Networks and are summarized in Table 3. From Table 3 we can see that one team had significant $P$-values for both the AUPR and AUROC. (For the other four teams, neither the AUPR nor the AUROC yielded significant $P$-values.)

To visually compare the performance of the submissions, we performed agglomerative unsupervised hierarchical clustering on the vectors $\text{prec}(k), k = 1, \ldots, 200$ (Fig. 10), the precision at depth $k$ in the submitted list. We only show the precision up to depth $k = 200$, rather than the full length of the prediction list (1,070 total predictions) to have better visualization of the precision at the most trusted predictions. None of the predictions clustered with the gold standard; rather, all predictions grouped together. However, Team 1 more consistently predicted correct interactions down to deeper entries in the prediction list. Interestingly, the first correct prediction of Team 1 was made only at depth 16 (taking away the pairs not considered in the gold standard). For Teams 3 and 4, the first correct prediction was the third and second, respectively. At their 50th prediction ($k = 50$), Team 1 had 12% of precision, whereas Teams 2 through 5 had precisions of 6%, 2%, 4%, and 6%, respectively.

The best performing team was (the team leader is underlined):

- Team 1: Hon Nian Chua, Willy Hugo, Guimei Liu, Xiaoli Li, Limsoon Wong, and See-Kiong Ng, from Singapore’s Institute for Infocomm Research, and the National University of Singapore, Singapore.
FIGURE 10. Heat map and hierarchical cluster analysis of the precision for the gold standard and the five submissions for the protein–protein interaction subnetwork challenge. The precision is shown down to the 200th position, out of the 1,070 possible entries in the prediction list. See Figure 9 legend for an explanation of the color scale. (Color is shown in online version.)

This team discusses its best performing strategy in the chapters dedicated to the best performing teams, later in this volume. The approach taken by the best performing team integrates seven protein–protein interaction prediction techniques, including domain–domain interactions, interaction motifs, paralogous interactions, Gene Ontology–based protein function similarity, and graph theoretic techniques. Using currently known physical protein–protein interactions from the BioGRID database as a training set, Team 1 trained each of their methods, and then integrated those methods into their prediction framework.

We asked the remaining teams to voluntarily share their methodologies with us, so that we could also extract lessons from the strategies that performed sub-optimally. Only Team 4 responded. Team 4 described a protein–protein interaction pair by a total of 28 features representing 17 distinct groups of biological data sources. These data sources could be divided into four categories: direct experimental data sets (two-hybrid screens and mass spectrometry), indirect high throughput data sets (gene expression, protein-DNA binding, biological function, biological process, protein localization, protein class, essentiality, etc.); sequence-based data sources (domain information, gene fusion, etc.), and features representing relationships between proteins on various graphs (e.g., 2-path on two-hybrid graph). They applied a “logistic regression” classifier to predict if a protein pair interacts or not. Their training set for this classification was derived from the BioGRID yeast PPI dataset.

It is hard to ascertain why Team 1 did better than Team 4. It is possible that the use of the full complement of protein–protein interactions from BioGRID used by Team 4, rather than the pruned out version (leaving out the nonphysical interactions) used by Team 1, may have harmed Team 4’s performance.

The Five-Gene Network Challenges

A total of 11 teams submitted predictions to five-gene network challenges, and each of the 12 categories received at least one
submission. The breakdown of participants per challenge is summarized in Table 4. The bulk of the submissions were in the FiveGeneNet1 (qPCR data) subchallenge. However, only two categories—those shaded in Table 4—had statistically significant scores. Interestingly, of the FiveGeneNet2 (chip data) subchallenge submissions, none of the participants made predictions for edges involving genes in the five gene network (recall that these 5 genes were hidden among the 588 genes whose gene expression data was provided); none of the submissions contained any of the network genes.

The details of the two categories that yielded significant predictions in this challenge (shaded in Table 4) are listed in Tables 5 and 6. Predictions were assessed as described in earlier sections. From Table 5 we can see that in the Undirected Unsigned category of the
FiveGeneNet1 (qPCR data) subchallenge, two of the four teams had statistically significant submissions. Even though Team 1 was clearly the best performing team, Team 2 was not that far behind. We can see some of these teams’ performance details in Figure 11A, depicting a hierarchical clustering of precision. While the second interaction predicted by Team 2 was not correct, the 3rd to 7th gene-to-gene connections were correct. Indeed, at depth $k = 7$, this team had a precision of 86%. However, Team 1 was more consistent in its performance, which gave it the edge.

The Directed Excitatory category of the FiveGeneNet1 (qPCR data) subchallenge had a clear best performing team, as can be seen in the heat map and dendrogram of Figure 11B depicting a hierarchical clustering of the precision. Team 0 correctly predicted the first three out of the 5 excitatory connections in this network and clustered tightly with the gold standard precision vector.

The best performing teams therefore were (team leaders are underlined):

- Team 0: Angela Baralla, Wieslaws Mentzen, and Alberto de la Fuente, from the CRS4 Bioinformatica, Università degli Studi di Sassari, Italy
- Team 1: Daniel Marbach, Claudio Mattiussi, and Dario Floreano, from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.

These teams discuss their best performing strategies in their respective chapters later in this volume. Even though the methods used by these two teams were different in many ways, it is interesting to highlight the important similarities that these methods share.

The approach taken by Team 1 was a very innovative one. They proposed a variant of
a genetic algorithm that encoded the logic of gene regulatory networks, thus integrating prior knowledge into the reverse engineering procedure itself. In this way, the search was biased toward biologically plausible solutions. They ran their Analog Genetic Encoding (AGE) algorithm 50 times and estimated the reliability of predicted edges by the frequency with which a particular connection occurred in the ensemble of predictions.

Team 0 modeled the kinetics of the system with a set of ordinary differential equations whose parameters represented the network connections. They determined the parameters using nonlinear optimization to fit the dynamics of their model to different combinations of the data provided in the challenge. Independent fits to these different combinations provided many results in an ensemble of predictions that were averaged to yield the final parameters from which the connections were inferred.

We asked the remaining teams to voluntarily share with us what their methodology was, so that we could also extract lessons from the strategies that performed sub-optimally. Three teams responded. One of the responding teams that participated in the Directed Unsigned/FiveGeneNet1 (qPCR data) category (summary tables not provided for this category), obtained AUPR and AUROC P-values of 0.99 and 0.88. They suggest that the poor performance of their method may have been due to the fact that they did not focus on pairwise connections. The Bayesian network with continuous variables method they used learns the best set of parents for each variable. Thus, if a variable has more than one parent, it is difficult to assign a confidence to the edge between that variable and each parent, independently of the other parents. Team 4 used Random Forests composed of 2,000 decision trees, ranking how informative to a particular gene were the other genes. This was done by fitting a regression model for each gene in terms of the values (past and present) of the other genes. Directions were resolved by using information theory criteria. Team 5 used the time course PCR data and applied a recursive L1 optimization to obtain a sparse model.

Let us point out an important difference in the comparison between the best performing teams and the other teams. The best performers modeled the five-gene network system using the knowledge that the system consisted of a gene regulatory network. Even though the details are different both best performers used kinetic models to represent the evolving system and found the best models that fit the data. The poorly performing methods, instead, used Bayes network inference and regression models and attempted to use probability theory to capture a causal kinetics that was probably better modeled by rate equations.

The in Silico Network Challenges

A total of 25 teams submitted predictions to the in silico network challenges. Not all of the 18 categories received submissions, however. The breakdown of participants per challenge is summarized in Table 7, where the number of teams that submitted in each category appears between parentheses, and each table entry shows the best performing team(s) if any.

The entries where there are two teams separated by a slash sign are the results of an interesting turn of events. After the submission deadline, Team 2 participants realized that they had a formatting error in their submission of directed networks. In the submission format, the source for each network edge had to be indicated in the first column and the destination in the second column. Team 2 submissions had the order inverted, causing the directionality of their edges to be the reverse from their intended directionality. That was not a problem for the undirected networks, where they obtained a remarkable score of 61.3 in the unsigned category. However, the mistake was devastating for the directed network submission. Team 2 noticed this mistake after the results and best performing teams had been announced, and they requested that we evaluate the same directed
TABLE 7. Summary of the participation to the different categories of the In Silico network challenges$^a$

<table>
<thead>
<tr>
<th>InSilico1</th>
<th>InSilico2</th>
<th>InSilico3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directed</td>
<td>Undirected</td>
<td>Directed</td>
</tr>
<tr>
<td>Unsigned</td>
<td>Excitatory</td>
<td>Inhibitory</td>
</tr>
<tr>
<td>(7) Team 2s/</td>
<td>Team 1</td>
<td>(6) Team 3</td>
</tr>
<tr>
<td>Team 1</td>
<td>Team 1</td>
<td>(2) Team 4</td>
</tr>
<tr>
<td>(8) Team 2s/</td>
<td>(6) Team 1</td>
<td>(1) non-sgnif</td>
</tr>
<tr>
<td>Team 2</td>
<td>Team 2</td>
<td>No Subm</td>
</tr>
</tbody>
</table>

$^a$The rows indicate the InSilico1, 2, and 3 subchallenges. Shaded cells are discussed further in the text. In each table entry, the number between parentheses indicates the number of participants in the corresponding category, and the text indicates either the best performing team(s), that no submission attained a significant score (non-sgnif), or that there were no submissions (No Subm).

TABLE 8. Performance of submissions to the InSilico1 network challenge, Directed Unsigned category$^a$

<table>
<thead>
<tr>
<th>Team</th>
<th>Precision at $n$th correct prediction</th>
<th>$P$-value</th>
<th>AUROC</th>
<th>$P$-value</th>
<th>AUROC</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>0.5 0.5 0.42 0.32 0.20</td>
<td>1.05 $10^{-16}$ 0.813 1.1 $10^{-22}$ 37.95</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 3</td>
<td>1 0.18 0.26 0.18 0.12</td>
<td>1.88 $10^{-9}$ 0.72 1.1 $10^{-12}$ 20.70</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 5</td>
<td>1 1 0.62 0.06 0.094</td>
<td>4.48 $10^{-6}$ 0.56 0.014 7.19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 6</td>
<td>0.5 0.25 0.36 0.12 0.085</td>
<td>3.24 $10^{-5}$ 0.57 7.3 $10^{-3}$ 6.63</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 7</td>
<td>0.5 0.4 0.13 0.067 0.057</td>
<td>0.011 0.53 0.12 2.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 8</td>
<td>0.02 0.026 0.033 0.038 0.038</td>
<td>0.72 0.49 0.57 0.39</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 2</td>
<td>0.018 0.033 0.037 0.036 0.038</td>
<td>0.72 0.49 0.65 0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 2s</td>
<td>1 1 1 1 1</td>
<td>1.58 $10^{-54}$ 0.82 3.06 $10^{-24}$ 77.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$^a$The columns are as in Table 2.

submissions they had sent, with the first and second columns swapped. This constituted a new (unofficial) entry, that we denoted Team 2s (s for swapped). Team 2s performed better than the official best performer in two of the directed categories for the InSilico1 challenge. However, Team 2s was not considered an official participant of DREAM2, and wherever Team 2s shares an entry with Team 1 in Table 7, Team 1 was the official best performer.

The Undirected Signed categories didn’t receive any submission, and the Undirected Unsigned received only four submissions in total, and only one of them was significant. It seems that, except for Team 2, teams with algorithms to predict directed networks were not interested in making predictions for the undirected categories.

The bulk of the submissions were in the directed categories for the InSilico1 and InSilico2 challenges. Even within the directed categories, participants seem to have been reluctant to participate in the more realistic and difficult InSilico3 challenge. However, there were submissions with significant scores in some of the InSilico3-directed categories.

The scores for all the teams in the Directed Unsigned categories in the InSilico1 and 2 challenges (those shaded in Table 7), are listed in Tables 8 and 9. Team 2s had an overall score of 77.3 in the InSilico1 Directed Unsigned category, and its first 33 predictions were actual edges in the InSilico1 network. Team 1 also had an excellent score of 37.95 for the same category, but its precision was not as impressive as that of Team 2s. Figure 12A shows a heat map of the precision as we go down in the list of predicted edges. Team 2s clearly clusters with the gold standard, showing an incredible consistency in its predictions. Team 1 shows also
TABLE 9. Performance of submissions to the InSilico2 network challenge, Directed Unsigned category

<table>
<thead>
<tr>
<th>Team</th>
<th>1st</th>
<th>2nd</th>
<th>5th</th>
<th>20th</th>
<th>AUROC</th>
<th>AUROC</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 3</td>
<td>1</td>
<td>0.67</td>
<td>0.71</td>
<td>0.46</td>
<td>0.26</td>
<td>2.64 $10^{-22}$</td>
<td>10$^{-22}$</td>
</tr>
<tr>
<td>Team 7</td>
<td>1</td>
<td>1</td>
<td>0.14</td>
<td>0.15</td>
<td>0.66</td>
<td>4.87 $10^{-12}$</td>
<td>6.6 $10^{-12}$</td>
</tr>
<tr>
<td>Team 5</td>
<td>0.5</td>
<td>0.67</td>
<td>0.56</td>
<td>0.11</td>
<td>0.11</td>
<td>5.6 $10^{-8}$</td>
<td>6.0 $10^{-8}$</td>
</tr>
<tr>
<td>Team 1</td>
<td>1</td>
<td>0.5</td>
<td>0.12</td>
<td>0.04</td>
<td>0.059</td>
<td>0.01</td>
<td>0.49</td>
</tr>
<tr>
<td>Team 2</td>
<td>0.14</td>
<td>0.25</td>
<td>0.077</td>
<td>0.045</td>
<td>0.049</td>
<td>0.084</td>
<td>0.513</td>
</tr>
<tr>
<td>Team 8</td>
<td>0.038</td>
<td>0.04</td>
<td>0.039</td>
<td>0.04</td>
<td>0.040</td>
<td>0.52</td>
<td>0.50</td>
</tr>
</tbody>
</table>

aThe columns are as in Table 2.

FIGURE 12. Heat map and hierarchical cluster analysis of the precision for the In Silico network challenge. (A) The gold standard and the eight submissions for the InSilico1 Directed Unsigned category. (B) The gold standard and the six submissions for the InSilico2 Directed Unsigned category. The precision is shown down to the 200th position, out of the 2,450 possible entries in the prediction list. See Figure 9 legend for an explanation of the color scale. (Color is shown in online version.)
a consistency in having its correct prediction toward the top of the list. (This can be seen in a tendency for the Team 1 precision to have a darker to red coloration in the top predictions.)

In the InSilico2 challenge, it was Team 3 that topped the scores. Interestingly, Teams 1 and 2 did not score as well in this challenge as in the InSilico1 challenge. Recall that the difference between these two challenges resides in the random topology of InSilico1 and the scale free topology of InSilico2. This is a clear demonstration of the influence of topology in the reconstruction of the networks, with some algorithms better suited to certain global topological structure.

Overall there were two teams, Team 1 and Team 2 (and its swapped version 2s) that had an excellent performance throughout all the categories. Team 3 performed very well in the InSilico2 Undirected Unsigned category, and Team 4 had a modest but nonetheless best performing score of 6.3 in the InSilico3, Directed Excitatory category.

The best performing teams were as follows (team leaders are underlined):

• Team 1: Alan Scheinine, Wieslawa Mentzen, G. Fotia, E. Pieroni, F. Maggio, G. Mancosu, and Alberto de la Fuente, CRS4 Bioinformatica, Italy
• Team 2 (and 2s): Mario Lauri, Francesco Iorio, and Diego di Bernardo, Telethon Institute of Genetics and Medicine, University of Naples “Federico II,” and University of Salerno, Italy.
• Team 3: Mika Gustafsson, Michael Hörnquist, Johan Björkegren, and Jesper Tegnér, from Linköping University and Karolinska Universitetssjukhuset, Sweden
• Team 4: Tejaswi Gowda, Sarma Vrudhul, and Seungchan Kim, from Arizona State University, Tempe, Arizona, and Translational Genomics Research Institute, Phoenix, Arizona.

These teams discuss their best performing strategies in their respective chapters later in this volume. Team 1 used a linearized dynamical system model under perturbations. For the InSilico1 and 3 challenges only the heterozygous knock-out data was considered. For the InSilico2 only the time-series data was used. For InSilico3, many edges were assumed to be nonexistent based on biological reasoning. In all cases a matrix had to be inverted to obtain the connectivity networks. Team 2 used their NIR (Network Identification by Reverse-engineering) method, based also on a linearized dynamical system model under perturbations, but with the constraint that each gene can be controlled by, at most, ten other genes. Like Team 1, Team 2 only used the heterozygous knock-out data. Team 3 also described the underlying system as a set of differential equations in which the dependence of a gene on another gene was mediated by a function chosen from a library of functional forms. The connectivity assignment, parameter estimation, and selection of nonlinear transfer functions were performed by least angle regression and cross-validation on all the available data. Team 4 modeled the interaction between each gene and its controlling genes in terms of a so-called threshold function. This is a function that assumes the value 1 if the linear combination of the levels of the controlling genes is above a threshold and 0 otherwise. The gene abundance was discretized into four levels. The weights of the linear combination were found by optimization against the time course data, and the connectivity was computed by adding only genes that reduce the errors in an iterative procedure.

We asked the remaining teams to voluntarily share their methodology with us, so that we could also extract lessons from the strategies that performed sub-optimally. Two teams responded. Team 7 used a method that assumed a nonlinear differential equation model of mRNA transcription and used regularized kernel regression to assess the degree to which a set of potential regulators is able to explain the expression of each gene. Note that Team 7 obtained quite a significant score for the InSilico2,
**TABLE 10. Summary of the participation to the different categories of the Genome Scale network challenges**

<table>
<thead>
<tr>
<th>Genome scale challenge</th>
<th>Directed unsigned</th>
<th>Directed excitatory</th>
<th>Directed inhibitory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular submission</td>
<td>(5) Team 1</td>
<td>(2) Team 2 – score: 9.6</td>
<td>(2) non-sgnif</td>
</tr>
<tr>
<td>w/Dimitris effect</td>
<td>N/A</td>
<td>(3) Team 1 – score: 26.4</td>
<td>(3) Team 1 – score 26.5</td>
</tr>
</tbody>
</table>

aThe row labeled “w/Dimitris effect” explores the result of using the Directed Unsigned submission of Team 1 as either Directed Excitatory, or Directed Inhibitory. The shaded cell is discussed further in the text. In each table entry, the number between parentheses indicates the number of participants in the corresponding category, and the text indicates either the best performing team, that no submission attained a significant score (non-sgnif), or the score if the best performing team is not discussed in Table 11.

Directed Unsigned category. Figure 12B also shows that the precision for the top 5 predicted interactions was 100%. A second team shared their method as well. It used a regression model on the time course data and applied a recursive L1 optimization to obtain a sparse model.

It seems that the Teams 1 and 2, which had the winning strategies, managed to capture the *in silico* dynamics with linear models and perturbations. The added sparsity condition of Team 2’s NIR algorithm seems to have favored the reconstruction of the random network topology, but not the scale free topology, presumably because it might not have captured the dynamics at the hub nodes.

**The Genome-Scale Network Challenge**

A total of six teams submitted predictions to the Genome Scale network challenges. The breakdown of participants per subchallenge is summarized in Table 10, where the number of teams that submitted in each category appears between parentheses, and each table entry shows the best performing team, if any. Each of the three categories in Table 10 had at least two submissions. The Directed Unsigned category (shaded in Table 10) was the most populated one, with five submissions.

The row labeled Regular Submission was the one announced to the participants. After the results of the predictions were posted, however, the best performer of the Directed Unsigned category, Team 1 (Prof. Dimitris Anastassiou), made a curious request. Given that he hadn’t submitted predictions for the signed submissions, he requested whether we could evaluate his unsigned submission as the predictions to be used for directed excitatory and directed inhibitory. After we did the evaluations, it turned out that Team 1 was the best performing team for the signed categories as well! The fact that an unsigned category submission performed very well for a signed category submission was sufficiently perplexing, to give it a name. We called it the “Dimitris effect,” for the originator of the request. The row “w/Dimitris effect” indicates the corresponding score.

For the regular submission, the Directed Excitatory category best performer was Team 2 with a score of 9.6 (see Table 10). Adding Team 1’s submission with the Dimitris effect brought the best performer score up to 26.4. Likewise, it can be seen in Table 10 that for the regular submission, the Directed-Inhibitory category had no significant prediction (out of the two submissions) with a score of 9.6. Again, scoring Team 1’s submission with the Dimitris effect brought the best performer score up to 26.5.

Table 11 shows the details of the submissions to the Directed Unsigned category. While the *P*-values for the AUROC are very small (and the *P*-values for the AUPR couldn’t be extrapolated, and therefore are upper bound based on sampling), the precision starts to degrade very fast after about the 300th prediction, even for Team 1, as can be seen in Figure 13. Also in Figure 13, we can see that Team 3 and Team 4 have a reasonable precision up to the first 150 predictions, after which the precision degrades.
TABLE 11. Performance of submissions to the Genome Scale network challenge, Directed Unsigned category$^a$

<table>
<thead>
<tr>
<th>Team</th>
<th>1st</th>
<th>2nd</th>
<th>5th</th>
<th>20th</th>
<th>100th</th>
<th>500th</th>
<th>AUPR</th>
<th>P-value</th>
<th>AUROC</th>
<th>AUROC</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>1</td>
<td>1</td>
<td>0.833</td>
<td>0.8</td>
<td>0.82</td>
<td>0.102</td>
<td>0.097</td>
<td>$&lt;10^{-5}$</td>
<td>0.61</td>
<td>$3.0 \times 10^{-36}$</td>
<td>40.5</td>
</tr>
<tr>
<td>Team 3</td>
<td>1</td>
<td>0.4</td>
<td>0.556</td>
<td>0.769</td>
<td>0.654</td>
<td>0.045</td>
<td>0.063</td>
<td>$&lt;10^{-5}$</td>
<td>0.575</td>
<td>$6.4 \times 10^{-21}$</td>
<td>25.2</td>
</tr>
<tr>
<td>Team 4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.907</td>
<td>0.699</td>
<td>0.069</td>
<td>0.08</td>
<td>$&lt;10^{-5}$</td>
<td>0.571</td>
<td>$8.8 \times 10^{-20}$</td>
<td>24.1</td>
</tr>
<tr>
<td>Team 5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.952</td>
<td>0.351</td>
<td>0.045</td>
<td>0.057</td>
<td>$&lt;10^{-5}$</td>
<td>0.56</td>
<td>$1.9 \times 10^{-14}$</td>
<td>18.7</td>
</tr>
<tr>
<td>Team 6</td>
<td>1</td>
<td>1</td>
<td>0.556</td>
<td>0.278</td>
<td>0.086</td>
<td>0.028</td>
<td>0.03</td>
<td>$&lt;10^{-5}$</td>
<td>0.53</td>
<td>$9.6 \times 10^{-6}$</td>
<td>10.0</td>
</tr>
</tbody>
</table>

$^a$The columns are as in Table 2.

considerably. Indeed, no team seems to have a significantly better than random precision after the 500th prediction. This indicates a need to prune out false positives. Figure 13 shows that the precision of Team 4 is better than that of Team 3, even though the latter scores better than the former in Table 11. This is due to the fact that the AUROC of Team 3 is marginally better that that of Team 4 and that we couldn’t properly score the precision in these submissions beyond the upper bound of $10^{-5}$. It is very likely that if our AUPR $P$-value had been more precise, Team 4 would have scored higher than Team 3, as suggested by its clustering with Team 1 in Figure 13.

The best performer team was (team leader is underlined):

- Team 1: John Watkinson, Kuo-ching Liang, Xiadong Wang, Tian Zheng, and Dimitris Anastassiou, all from Columbia University.

This team describes its best performing strategy in a chapter later in this volume. The strategy of Team 1 was based on the context likelihood of relatedness (CLR) algorithm.
enlarging it with additional complementary information using the information theoretic measure of synergy (which makes use of three variable associations) and assigning a score to each ordered pair of genes measuring the degree of confidence that the first gene regulates the second.

We asked the remaining teams to voluntarily share with us what their methodology was, so that we can also extract lessons from the strategies that performed sub-optimally. Team 3, whose performance was the second best, responded. They used their LARS (a linear regression with a sparsity-promoting prior/penalty term) algorithm, and used a regularization to minimize the number of genes regulated by each transcription factor.

A take home lesson is that the measure of synergy used by Team 1, which goes beyond the pairwise mutual information to include the state of a third gene that might enhance the interaction between the pair, seems to have given Team 1 an extra push to achieve the best performance in this challenge.

Conclusions

The lessons learned in DREAM2 can be divided into two classes: lessons for future development of DREAM challenges, and lessons on the current state of affairs of reverse engineering methods. We will address the former class first.

At the end of the DREAM2 conference, held at the New York Academy of Sciences on December 2 and 3, 2007, where the results from the challenges were discussed, there was a lively discussion where a number of different ideas were suggested.

An important, if somehow philosophical, comment was whether we should predict a network model, which is a concept rather than an observation, or a measurement related to an observation (e.g., gene expression or fluorescence intensity). In the same vein as the equations of mathematical-physics do not “hide” within the physical phenomena they describe, the networks we proposed as gold standard models for inference in the DREAM2 challenges do not truly “exist” in nature. Networks are a conceptual representation of cellular interactions. By giving an element taken from the realm of ideas an absolute reality, we are upholding the tenets of the Platonic idealistic philosophy. On the other side is the positivistic philosophy, which postulates that scientific knowledge can only be based on actual measurements. In other words, the positivist would posit that instead of predicting networks, DREAM challenges should be about predicting experimental results. This is an interesting discussion that makes the point that the original emphasis on network inference may be excessive. Future DREAM challenges will thus de-emphasize network inference in favor of methods that may use networks (or other models) to make predictions about observations that may be experimentally measured.

Another point of discussion was about the anonymity of the teams. There are pros and cons to disclosing the identity of all teams, but we prefer to stick to the Hypocratic dictum: do no harm. It is possible that revealing the identity of poorly performing teams could hurt its members in different ways (e.g., funding, tenure, or professional credibility). Thus, some potential participants may be deterred from participating because of the possible repercussions. What we need to know, in any case, is what method that team used, to be aware of its limitations. There are ways to disclose a methodology without disclosing its proponents, but the discussion is still open.

Some people felt that the design of our challenges ignored years and years of molecular biology. These critics propose to create challenges whose outcomes are biological predictions that extend and supersede what is already known. We believe that some challenges actually did that. For example, the BCL6 challenge asked for targets that are not in the literature and the five-gene synthetic network was recently accepted in the journal Cell. However the main
objective of the DREAM challenges is to assess the ability of our methods to do what we claim they do, for example, to find the hidden network, or to predict measurable results. Notwithstanding, it may be possible to create challenges that may both address the DREAM aims and generate novel biological hypotheses, emphasizing those predictions that have the greatest consensus among participants. We are working on it, but new ideas will be welcomed.

At DREAM2 some colleagues noted that calling ourselves “reverse engineers” implies that there is an “engineer” at work designing the biological systems. In the preface to a previous volume (“Reverse Engineering Biological Networks, Annals of the NYAS, volume 1115”), we explained why we chose those terms:

In the process of reverse engineering biological circuits, we are trying to “read the mind” of the unintelligent designer (evolution) that engineered the system. . . . We favor the terms reverse engineering because the knowledge it generates can then be used to forward engineer biological systems (i.e. to design novel working biological systems from first principles), which constitutes the basis for the budding field of synthetic biology.

We also confess that we like the “ring” of reverse engineering.

There were comments to the effect that the number of categories in Challenge 3, 4, and 5 should be reduced. As Alberto de la Fuente put it in the DREAM discussion forum:8 “Gene networks are directed networks, so we should infer directed networks; there is no need for an undirected category.” And then he added “there should be one overall signed category rather than decomposing into excitatory and inhibitory.” The reason we decomposed it in excitatory and inhibitory categories was to make the evaluation over a binary prediction. But we agree that in this case less is more, and future challenges will have fewer categories.

Let us now focus on the lessons learned from the submissions to the challenges. One important observation can be made from the BCL6 challenge submissions: of the 11 participating teams, the prediction of 6 of them was no better than a coin toss. This conclusion is worrisome. It means that many researchers who are experts in target identification may base their conclusions on potentially wrong premises. We are pleased that the team leader of Team 6 in Table 1 expressed appreciation for how much they had learned from having failed to produce a significant prediction in Challenge 1. This is an important call for caution. On the brighter side, we believe that the three best performing teams showed proof of principle that it is possible to infer transcripational targets accurately. It is very interesting to note that the aggregate predictions of these three teams performed much better than even the best of them. To aggregate the three team predictions, we did the following. Each team gave each presumed target a rank in the prediction list. For each gene, we summed the ranks of the three teams, yielding a list of genes with new ranks. Reordering this list according to the rank sum, we obtained a new prediction. The results were fantastic. The best of the three AUPR was 0.72, with a $P$-value of $1.4 \times 10^{-12}$; the new prediction yielded an AUPR of 0.85 and a $P$-value of $3.85 \times 10^{-17}$; the best of the three AUROC was 0.85, with a $P$-value of $3.3 \times 10^{-14}$; the new prediction yielded an AUROC of 0.93 and a $P$-value of $3.61 \times 10^{-20}$; the best of the three overall scores was 24.99; the new prediction yielded a score of 35.86; finally, the team with the largest run of correct predictions at the top of the list had 10 correct predictions before the first wrong prediction; the new prediction had 19 correct predictions before the first wrong predictions. Clearly, this shows that the “intelligence” behind the three methods can be easily aggregated to produce an integrative approach that is better than any method in isolation. This calls for a reverse-engineering-by-consensus approach that has not yet been explored by the community.

The upbeat message arising from transcriptional predictions, unfortunately, does not translate to the protein–protein interaction challenge. Here, four out of five participating teams had non-significant scores (i.e., no better than random). Even the best performing team, while statistically significant, had a
disappointingly low score. This is probably lit-
tle surprise to the field of protein–protein in-
teraction inference, which has struggled with
a high rate of both false positive and false
negative errors arising from the technology in-
volved. These may contaminate both the chal-
lenge gold standards and the training sets used
by the participants; or else, the computational
methods are not yet ready for prime-time in this
field. Most likely, poor results arise from a com-

bination of the two. In particular, we feel that
false negative errors from high-throughput ex-
perimental techniques may still be too high to
produce accurate gold standard negative sets.
Additionally, protein–protein interactions may
provide fewer clues for computational methods.
In any case, it seems that we still have much to
learn in the area of protein–protein interaction
prediction.

There were many non-significant categories
in the five-gene network challenge, and the
scores of those who performed best were mod-
est. It is surprising that a simple network of five
genes could be so hard to reconstruct. However,
the qPCR data available for the reconstruction
was scant and probably noisy. M. P. Cosma,
D. diBernardo, and collaborators have recently
written an interesting paper on this synthetic
biology network, with more data of better qual-
ity. It would be interesting for the teams that
participated in this challenge to revisit it with
the the new data. In the chip-data category, the
challenge was akin to finding a needle (the five
gene network) in a haystack (other 583 genes),
and there was no inferred connection that con-
tained any of the genes in question. However,
there were too few teams in this challenge to
extract any meaningful conclusion.

InSilico1 and InSilico2 network prediction
challenges had high participation, but not so
the more realistic InSilico3. It seems to be the
case that there are fewer methods that deal with
the more complicated and realistic case that the
InSilico3 challenge presents. This is clearly
a niche that needs to be filled. One clear lesson
learned from Challenge 4: one size doesn’t fit
all. Topology is very important in the ability
of a method to reconstruct the network, as the
team that did so well in the Erdos-Renyi topol-
ogy, performed very poorly in the scale-free
topology network. Challenge 4 Team 2 clearly
shone in the InSilico1 challenge, most likely be-
cause the data fit the assumptions of the algo-

rithm. This is a good lesson to learn. However
elegant an algorithm is, it needs to be designed
using all the biological knowledge of the system
available. Statistical methods of reconstruction
faired less well than mechanistic methods us-
ing rate equations, probably because the latter
were a better match to the nature of the data.

The genome scale challenge can be seen
under two lenses. On the one hand, it is en-
couraging that of the five teams participating
in the Undirected Unsigned category, all ob-
tained significant scores. On the other hand,
Figure 13 shows that even the best performer
is only scratching the surface when it comes to
precision: precision downgraded close to ran-
dom levels after the first 300 interaction pre-
dictions a mere 10% of the 3,102 true posi-
tive edges. Most likely, some gene interactions
can be dissected only under context-specific
conditions. Thus, the imprecision cannot be
completely accounted for by the algorithms. In
other words, the gold standard was much richer
than the experimental sets provided for the in-
ference. A curious phenomenon discovered was
the “Dimitris effect.” The prediction of Team
1, which was designed for unsigned predictions
without any attempt to infer signs of the con-
nection, faired better in the signed categories
than the methods created with the intention of
discovering the sign of the interaction. We do
not fully understand why this was the case. But
it is very likely that the ability of Team 1 to pre-
dict transcription factor-target interactions was
better than that of Team 2, thus compensating
for the lack of consideration for the signs in the
method.

There is much to mine in the submitted
data that also requires a community effort.
All the predictions from all the teams will
be posted in the DREAM website (http://
wiki.c2b2.columbia.edu/dream/) and will be
publicly available. It is plausible that a meta-analysis of the whole prediction set beyond what we did in this chapter might provide us with a few additional ideas to improve our algorithms. As much as the “reverse engineers” created their methods of network inference, those of us who are in the business of creating well-posed challenges have a long way to go, and many more lessons to learn. We are clearly in a time when methods proliferate with some fairing well and some fairing rather poorly. So the question remains: When it comes to reverse engineering biological circuits, is the glass half full or half empty?

Acknowledgments

Many people contributed to the DREAM project and DREAM2, in particular. Thanks to J. Jeremy Rice, with whom GS first dreamt of DREAM on a train trip. Thanks also to Fritz Roth, Joel Bader, Peicheng Du, Guillermo Cecchi, and Ajay Royyuru for fruitful discussions, and Dan Gallahan for his encouragement. We are grateful to the labs of Andrea Califano, Riccardo Dalla Favera, Marc Vidal, Maria Pia Cosma, Pedro Mendes, and Tim Gardener who gracefully provided the data for the challenges. We are indebted to Kai Wang, Adam Margolin, Haiyuan Yu, Diego diBernardo, Irene Cantone, and Boris Hayete for invaluable help in the curation for the challenges. The DREAM project would not exist without the teams, many of them anonymous, participating in the challenges. They put time and brain to solve the reverse engineering problems. We hope the fruit of their efforts will benefit that whole reverse engineering community. Financial support for DREAM from the IBM Computational Biology Center and the NIH Roadmap Initiative, through the Columbia University Center for Multi-scale Analysis Genomic and Cellular Networks (MAGNet) is acknowledged.

Conflicts of Interest

The authors declare no conflicts of interest.

Appendix

Precision Versus Recall Curve Beyond the Maximum Recall Value of a Set of Predictions

Suppose we have a sample space with $T$ elements (in our case, possible edges between nodes in a graph). These elements belong to one of two classes: the class of positive examples (edges of the graph), with $P$ elements, and the class of negative examples (edges absent from the graph), with $N$ elements. Clearly $T = P + N$. The fact that we know the class to which each example belongs makes this sample space a gold-standard.

A prediction is made in the form of an ordered list of $L$ samples taken from our sample space. This list is ordered such that the examples at the top of the list are those for which we have the higher confidence to be in the positive class. We will assume that the whole list with $L$ entries were to be taken as a prediction of the Positive set predictions. Clearly $L = TP_L + FP_L$. We now add, in random order, the remaining $T-L$ samples (on which no prediction was made) to the bottom of the original list with $L$ examples. We want to compute the precision and recall corresponding to the prediction that the $k$ ($k > L$) first samples in the resulting list are positive.

We first estimate the number of $TP$ and $FP$ resulting from adding $k-L$ random examples. These examples were randomly sampled from a set of $P – TP_L$ positives and $N – FP_L$ negatives. Therefore the probability $\rho$ of any of those $k-L$ examples to be a positive example is $\rho = \frac{P – TP_L}{T – L}$, and the probability of having gained $\Delta TP$ true positives in the added samples (between samples $L$ and $k$) follows a binomial distribution with parameters $k-L$ and probability $\rho$, whose average is $\langle \Delta TP \rangle = (k – L)\rho$. The precision $p(k)$ and recall $r(k)$ at depth $k$ in the list of predictions, therefore, are functions of the
random variable $\Delta TP$:

$$p(k) = \frac{TP_L + \Delta TP}{k}; \quad r(k) = \frac{TP_L + \Delta TP}{P}. \tag{A1}$$

It follows that the average precision and recall at depth $k$ are:

$$\langle p \rangle(k) = \frac{TP_L + (k - L)p}{k}; \tag{A2}$$

$$\langle r \rangle(k) = \frac{TP_L + (k - L)p}{P}. \tag{A3}$$

By allowing $k$ to span the range between $L$ and $T$, the previous two equations give us the shape of the precision and recall curve, parameterized by $k$.

We can also write down an explicit function of precision versus recall curve. Solving for $k$ in Equation (A3) and replacing in Equation (A2) we obtain

$$\langle p \rangle = \frac{P\langle r \rangle}{P\langle r \rangle - r_L} + L\rho, \tag{A4}$$

where $r_L = TP_L/P$ is the recall at depth $L$.

Note that if $L = 0$ (no predictions submitted), then $r_L = 0$, and $\langle p \rangle = \frac{P}{T}$, that is, it’s a constant. In other words the average precision recall curve for a random prediction list is a constant and does not depend on the recall. In this case the integral of the area under the precision versus recall curve (or AUPR for short, to be notated $\mathcal{A}$) is $\mathcal{A} = \frac{P}{T}$.

In the general case, Equation (A4) shows that the precision versus recall curve, after the addition of $T-L$ random predictions at the end of the original $L$ predictions, has a hyperbolic shape and goes through the points $(r_L, p_L)$ (where $p_L = TP_L/k$ is the precision at depth $L$) and $(1, P/T)$.

The slope of the curve for recall greater than $r_L$ is

$$\frac{d\langle p \rangle}{d\langle r \rangle} = \left(1 - \frac{TP_L}{L}\right)$$

$$\times \frac{\rho PTL}{(T - L)\left(L\rho + P\langle r \rangle - r_L\right)^2},$$

which clearly is negative (decreasing curve) if the precision at recall $r_L$ is larger than random $(P/T)$, or positive (increasing curve) if it is smaller than random. The added area under the curve, that is, the area beyond the precision recall curve above recall $r_L$ is

$$\mathcal{A} = \int_{r_L}^{1} \langle p \rangle dr = \rho(1 - r_L)$$

$$+ \rho\left(r_L - \frac{L\rho}{P}\right)\ln\left(\frac{L\rho + P(1 - r_L)}{L\rho}\right).$$

How to Integrate the Precision Versus Recall Curve to Find the AUC

A practical issue to address is how to interpolate between points in the precision versus recall curve when we try to compute the AUPR. Once we have decided on the appropriate interpolation, it will be straightforward to proceed with the integration of the curve. Each time we go down the list of predictions from item $k$ to item $k + 1$, two things can happen:

(a) We gain a $FP: FP_{k+1} = FP_k + 1$.

In this case, the number of $TP$'s didn’t change $(TP_{k+1} = TP_k)$, and the PR curve moved from point $k$ at $(\frac{TP_k}{P}, \frac{TP_k}{TP_k + FP_k})$ to point $k + 1$ at $(\frac{TP_k}{P}, \frac{TP_k}{TP_k + FP_k + 1})$. The area under the curve therefore didn’t change: $\mathcal{A}_{k+1} = \mathcal{A}_k$.

(b) We gain a $TP: TP_{k+1} = TP_k + 1$.

In this case, the number of $FP$'s didn’t change $(FP_{k+1} = FP_k)$, and the PR curve moved from point $k$ at $(\frac{TP_k}{P}, \frac{TP_k}{TP_k + FP_k})$ to point $k + 1$ at $(\frac{TP_k}{P}, \frac{TP_k}{TP_k + FP_k + 1})$. Assume that there is a continuous change in the number of $TP$’s from $TP_k$ to $TP_{k} + 1$. Call such continuous variable $x$. Between point $k$ and point $k + 1$, we traverse the curve $(\frac{TP_k}{P}, \frac{TP_k + 1}{TP_k + FP_k + 1})$ for $x$ between 0 and 1.

Calling $r(x) = (TP + xP)/P$ (as in recall), we see that the PR curve can be parameterized by $r$ as $(r_k, \frac{rP}{TP_{k+1}})$ for $r$ between $TP_k/P$ and
FIGURE A1. Pseudo-code to determine the precision versus recall, as well as the ROC curves of a set of complete predictions \( L = T \) or possible incomplete predictions \( L < T \). (Color is shown in online version.)

\[
\frac{TP_k + 1}{P} \cdot \int_{TP_k/P}^{(TP_{k+1} + 1)/P} \frac{rP}{rP + FP_k} \, dr
\]

\[
= \mathcal{A}_k + \frac{1}{P} \left( 1 - FP_k \ln \left( \frac{k + 1}{k} \right) \right).
\]

Note that if the interpolation between point \( k \) and point \( k + 1 \) were to be done linearly instead of hyperbolically, the resulting areas \( \mathcal{A}_k \) would be

\[
\mathcal{A}_{k+1} = \mathcal{A}_k + \frac{1}{P} \left( 1 - FP_k \frac{k}{k(k + 1)} \right) \quad \text{(linear interpolation case)}.
\]

It is then clear that a linear interpolation when we integrate would result in a more optimistic AUPR than the hyperbolic and more accurate functional form.

### Determination of the Prediction Accuracy for DREAM2

Given the considerations of the previous sections of this Appendix, the pseudo-code to determine the precision versus recall, as well as the ROC curves of a set of (possibly) incomplete predictions (i.e., \( L < T \)) is as shown in Figure A1.

### References


