Gene Regulatory Network Reconstruction Using Bayesian Networks, the Dantzig Selector, the Lasso and Their Meta-Analysis

Matthieu Vignes*, Jimmy Vandel, David Allouche, Nidal Ramadan-Alban, Christine Cierco-Ayrolles, Thomas Schiex, Brigitte Mangin, Simon de Givry

SaAB Team/BIA Unit, INRA Toulouse, Castanet-Tolosan, France

Abstract

Modern technologies and especially next generation sequencing facilities are giving a cheaper access to genotype and genomic data measured on the same sample at once. This creates an ideal situation for multifactorial experiments designed to infer gene regulatory networks. The fifth "Dialogue for Reverse Engineering Assessments and Methods" (DREAM5) challenges are aimed at assessing methods and associated algorithms devoted to the inference of biological networks. Challenge 3 on "Systems Genetics" proposed to infer causal gene regulatory networks from different genetical genomics data sets. We investigated a wide panel of methods ranging from Bayesian networks to penalised linear regressions to analyse such data, and proposed a simple yet very powerful meta-analysis, which combines these inference methods. We present results of the Challenge as well as more in-depth analysis of predicted networks in terms of structure and reliability. The developed meta-analysis was ranked first among the 16 teams participating in Challenge 3A. It paves the way for future extensions of our inference method and more accurate gene network estimates in the context of genetical genomics.

Introduction

Inferring gene regulatory networks

Gene regulatory networks (GRN) are simplified representations of mechanisms that make up the functioning of an organism under given conditions. A node in such a network stands for a gene *i.e.* a DNA fragment that encodes a functional agent of the cell such as a protein. Proteins are among the most well-studied acting protagonists in living organisms. In large part, their synthesis is effectively regulated by other interacting proteins. In a GRN, edges depict causal relationships between sources and targets for gene activities. Hence a convenient representation for GRNs are directed graphs. The objective of the third DREAM5 challenge was to infer causal relationships in artificial complex networks.

More generally, a biological network is defined by constituents at different levels, such as DNA sequences, RNAs (messengers, but also small RNAs), proteins, metabolites. Discounting epigenetic effects, genes barely interact directly. They are rather activated or repressed through the action of specific components acting at other scales [1]. The work presented in this paper is focused on gene regulatory interactions. This representation maps the action of all cellular components onto gene space. Potential applications still benefit from this simplified interpretation of the complex system. For example, the search of candidate genes that target changes in a phenotype of interest [2], the study of evolutionary aspects of biological networks [3,4] so as to link their structure to

functional properties [5,6] all use the representation of gene regulatory networks.

Initially, specific attention has been devoted to understanding the dynamics [7] and principles governing gene regulation, using either the first rules in logic to capture the absence or presence of cycles in a Boolean formalisation of a GRN [8]. Later, [3,9] also studied the successive refinements on gene network topology and their functional consequences. In the past ten years, motivated by the abundance of micro-arrays, a huge effort has been devoted to GRN inference. The methods that were proposed and developed include analyses based on correlations in the data [10], systems of ordinary or partial differential equations that give a plausible physico-chemical modelling [11], systems of linear equations [12] and Bayesian networks (BN, [13]) to cite only a few. Additional improvements were proposed depending on the exact nature of the data at hand (*e.g.* time series, [14–16]) and biological *a priori* knowledge [17].

Genetical genomics paradigm for gene network inference

In order to decipher causal links, the above-cited methods rely on expensive and still technically challenging time series data or on many experiments perturbing the systems from a steady state (*e.g.* by studying the effect of knocking out a gene). "An ideal experimental design for causal inference is randomised, multifactorial perturbation" recalls the website of the third challenge of the

Dialogue for Reverse Engineering Assessments and Methods (DREAM5, [18]) giving a makeover to Fisher's work on experimental design [19] in molecular biology data analysis. Genetic polymorphisms in a segregating population are ideal settings for multifactorial perturbations of a living system: each allele is a potential source of perturbation for network behaviour. Recombination and segregation events that occur during genetic crosses, randomise the distribution of these alleles among the lines derived from two genetically known and diverse parents. Systems genetics, or more precisely "genetical genomics" [20,21], is the study of how such randomised genetic perturbations can directly or indirectly affect numerous complex traits. These traits can be either qualitative phenotypes of interest or quantitative measurements reflecting the activity of cells like transcriptomics data. The variety of patterns in trait responses on genotyped individuals in the segregating population are used to draw causal inference. The added value of having both genetic and perturbed phenotypic (expression) data has already been demonstrated, in particular to infer causality [22]. Existing works that elucidate GRN structure based on genetical genomics data have been using Bayesian networks (BN) using genetic data as prior information [23] or multivariate regression in a structural equation modelling (SEM) framework with multiple testing and greedy search steps [24]. Their efficiency and accuracy in dealing with high dimensional data set is still very limited.

In this paper, we consider complementary approaches that could potentially improve over state-of-the-art methodologies to perform GRN inference from systems genetics data sets, namely (i) penalised regressions: Lasso [25]) and the Dantzig Selector [26] which seek linear interpretable dependencies with a controlled level of parsimony and (ii) BN with an appropriate scoring function and an integrated treatment of genetic and genomic data. These approaches are used as inputs to feed a consensus statistical meta-analysis approach that combines the best of other learning algorithm results, and which emerged as the best performer for the DREAM5 Challenge 3A on GRN inference in systems genetics. Since there is still no large experimental data set for which a reliably known large size gene network exists, the challenge offered simulated data based on differential equation simulation, defining Gold Standard data sets.

The first section of the paper is devoted to the results we obtained. A discussion on the relative merits and limitations of the proposed methods follows. The "Material and Methods" section precisely describes the data and the different methods used, including specific adaptation to the data sets and post-processing used to produce network estimates.

Results

In this section, we present our results and the prediction performance achieved according to the DREAM5 challenge criteria and we then give a more in-depth analysis in order to gain more insight on learnt GRNs. The data sets provided by the challenge organisers, which are described in more detail in the "Material and Methods" section, contained simulated genotypes in recombinant inbred line (RIL) populations of variable size (100, 300 or 999 individuals) and their associated expression levels, which were governed by inductive or repressive effects of genes on each other according to the topology of plausible networks to recover. For each RIL population size, five networks with an increasing number of edges were simulated, so a total of 15 data sets were provided.

Since the meta-analysis that we used was the best performer of the challenge, we focus on the results obtained using this consensus

method. To illustrate the complementary of the different methods (BN, Lasso and Dantzig-based regressions) that supplied input edges to the meta-analysis, we also present several aspects of their predictions. According to DREAM5 specifications, a predicted network topology is defined by a list of directed edges ranked according to a non-increasing order of confidence score.

General results

Edge lists were compared both to (i) Gold Standard files, namely the correct list of edges used in simulated models and to (ii) the pool of all edges that were submitted by other participating teams.

Receiver Operating Curves (ROC *i.e.* True positive vs. False positive rates – FPR) and precision vs. recall (PR) curves were produced for each network. The false positive rate assesses the trend of the method to produce incorrect edges. The recall is equal to the true positive rate and measures the power of a method to recover the complete set of true edges. The precision (the rate of correctly made predictions) is an indicator of the reliability of the predictions. Curves obtained by the meta-analysis, BN, Lasso and Dantzig approaches on the sparsest network, with 999 individuals (Network1-A999), are shown in Figure 1.

In ROC curves, the point FPR = 0, recall = 1 corresponds to an ideal situation where all and only correct edges would be predicted. This ideal point was not reached by the meta-analysis, however an interesting trade-off value of (FPR = 0.05, recall = 0.8) was reached. The excellent FPR value was not surprising given that the simulated networks were rather sparse. The steep slope at the beginning of the ROC curve is a good indicator that the edge ranking produced by the methods was reliable.

The PR curve plots the ability of the methods to produce both reliable and comprehensive predictions. For example, with the 1,000 first edges obtained by the meta-analysis, a precision value of 75% means that three out of four inferred edges were correct, whilst the recall value of 25% means that one out of four original edges was recovered.

It should be noted that since the Lasso and Dantzig approaches produced up to 100,000 edges, so did the meta-analysis. The BN method produced sparser edge lists: between 2,900 and 5,800 edges per network. This makes the reliability of the score assigned to edges a key point: no one is really interested in predicting a true edge that is ranked 50,000th. In the next section, detailed features about our results are presented, and the emphasis is put on how these features can serve GRN inference when a Gold Standard network is unknown.

Table 1 presents the area under the ROC and PR curves (AUC) of the 15 inferred networks for the meta-analysis approach. Results clearly showed that reducing the size of the sample made the problem much harder. At the same time, it also appeared that increasing the edge density of the simulated network (from Network '1' with ~2,000 edges to Network '5' with ~5,000 edges) also made the challenge of GRN inference slightly more difficult, since prediction performances decreased.

Since the publication of the official results of the DREAM5 challenges, we have slightly improved the post-processing of our approaches. For example, the handling of edge direction is now identically dealt with by the two penalised regression approaches. Consequently, the meta-analysis AUC also changed. PR trade-offs were noticeably improved, whilst ROC slightly decreased.

The prediction of every method against the pool of all the predictions submitted by the teams that entered the challenge was also assessed. It was used to produce empirical *p*-value derived scores [27] that reflect how good each method performed in comparison to others and was eventually used to rank teams. Our meta-analysis method achieved first place in the three sub-
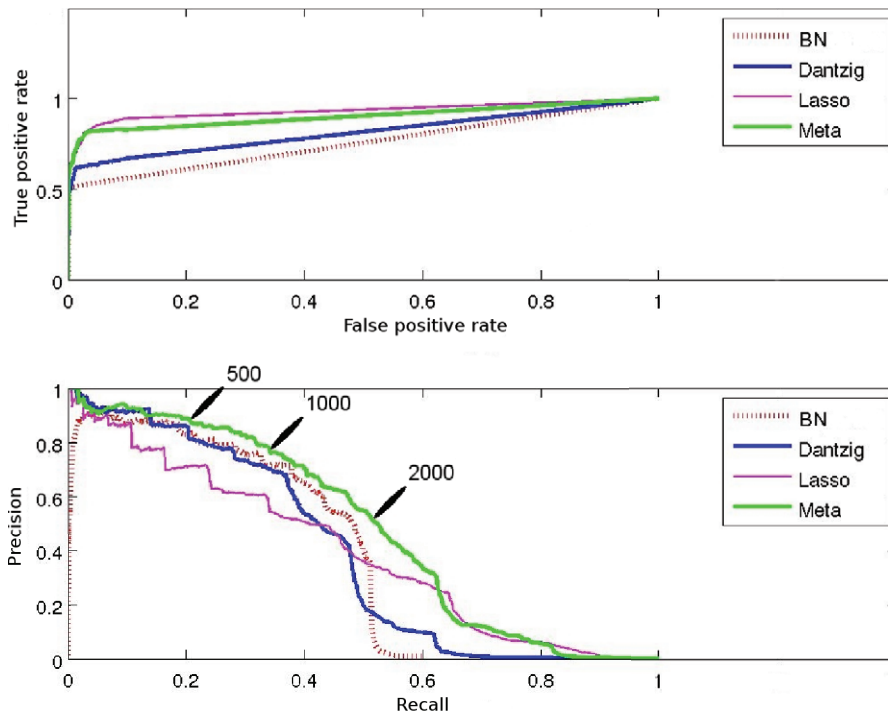
Figure 1. Accuracy results for the GRN inference methods: ROC curves (upper panel) and PR curves (lower panel) for Network1-A999. Meta-analysis: green, BN: dashed red, Lasso: purple and Dantzig: blue. Points for inferred networks of 500, 1,000 and 2,000 edges are indicated.
doi:10.1371/journal.pone.0029165.g001

challenges 3A (999, 300, 100 individuals) with respective overall scores of 140.6, 89.4 and 81.9. In sub-challenge 3A999, our scores were the best for both ROC and PR scores. In sub-challenge 3A300, two different teams provided better results: one for the ROC curve and one for the PR score, although none of them achieved a better overall score than our meta-analysis.

Detailed results

In this section, we present a detailed analysis of the results obtained on the most favourable case, which is Network 1 with 999 individuals (Network1-A999). This choice is naturally arguable, since a common situation in a systems genetics context is to infer relationships between genes when sample size is limited. It however gives an upper bound on the performances achieved on all networks and defines an ideal situation where the most reliable observations and conclusions can be drawn.

Correct edges come first. Since predicted edge lists can be as long as 100,000 edges and since we are interested in obtaining reliable and interpretable predictions only, we focus on the first 500, 1,000, 2,000 and sometimes 5,000 edges. The 'Results' section established that such short-list of predicted interactions simultaneously gave reasonably good coverages and acceptable precision levels (see corresponding precision and recall values in Figure 1). Moreover, they represent sets of edges whose sizes are reasonably manageable in the context of a 1,000 gene regulatory network that must be deciphered without any prior knowledge.

We tried to infer the directed network topology from the 500 first edges of the meta-analysis. 434 of them (86.8%) were correct, but 1,614 edges among the 2,048 edges of the true network were missing. So the recall was only 21.2%. When we used the 1,000 or 2,000 first edges, the recall increased to respectively 36.4% and 51.1% but the price to pay was a drop in precision to respectively

Table 1. AUC of the DREAM5 Challenge 3A for the meta-analysis of the SAaB team (source: [18]).

DREAM challenge		Area Under the Curve (AUC)				
		Network '1'	Network '2'	Network '3'	Network '4'	Network '5'
A999	PR	0.358[a]/0.482[b]	0.258/0.364	0.195/0.292	0.183/0.260	0.178/0.244
	ROC	0.933/0.902	0.885/0.845	0.844/0.808	0.821/0.784	0.813/0.768
A300	PR	0.211/0.248	0.144/0.175	0.141/0.159	0.132/0.141	0.113/0.131
	ROC	0.855/0.845	0.793/0.779	0.786/0.774	0.759/0.739	0.737/0.719
A100	PR	0.085/0.074	0.060/0.054	0.053/0.045	0.054/0.046	0.054/0.044
	ROC	0.754/0.750	0.718/0.713	0.696/0.694	0.676/0.671	0.670/0.666

[a]AUC official values issued by the DREAM organisers.
[b]AUC after minor corrections in our implementations.
doi:10.1371/journal.pone.0029165.t001

74.6% and 52.4%. So inferring half of the true network led to an inference noise of nearly 50%.

For denser networks, the precision remained the same, but the recall decreased since the total number of edges to predict was greater. As an example, in Network 5 (5,545 regulatory relationships), the 2,000 first edges produced by the meta-analysis had a precision rate of 54.2% but the total network coverage was only 20%. In this case, raising the total number of edges to keep for inference purpose was not a good option since using the 5,000 first edges indeed slightly increased the recall to 31.3%, but the precision then went down to 34.7% (data not shown).

In/Out-degree distributions. If initially the only knowledge that was available on the simulated networks was that they had a modular structure, the organisers later revealed how they were generated, including simulated in/out degree distributions. It is therefore informative to compare Gold Standard networks to predicted networks in terms of node degree distributions. Figures 2 and 3 compare plots of respectively in- and out-degree distributions for the true Network1-A999 and for networks inferred from the first 500, 1,000 and 2,000 edges predicted by the meta-analysis. The first result was that the larger the set of edges, the more accurate the predicted network topology: inferred degree distributions got closer to the correct ones when the edge list was increased. This would obviously not be true if we had considered much longer edge lists (which have poorer precision levels): a list of tens of thousands of edges would give too high a network connectivity, and degree distributions would be skewed. With the number of edges that we considered, distributions were skewed towards 0 and some nodes were

isolated even when 2,000 edges were kept (see the paragraph on largest connected components, below).

The in- and out-degree distributions of the true network and its modularity are global essential features of its topology. This modular structure appeared with as few as the first 500 edges and was clearly visible with the first 2,000 edges, as it is illustrated in Figure 4. However, the method had difficulties in locally capturing relationships for a node that had many incoming links: it was quite difficult to unmask regulatory hubs. For example, the true network had a dozen genes with more than 7 incoming edges and our predictions among the first 2,000 edges revealed only one node with 5 incoming links. Moreover, the true network had one 57-outgoing relationship hub and using the 2,000 first meta-analysis edges, we predicted only 30 such links for this hub. A consequence of this difficulty in predicting highly connected nodes was that our predictions overestimated the number of nodes with few regulatory connections.

Despite this, the meta-analysis performed relatively well at inferring networks with relatively accurate in- and especially out-degree distributions. In real biological data sets applications, if one had some prior knowledge about the true degree distributions (*e.g.* from another well-studied organism) plotting inferred node degree distributions would probably be a good tool for assessing network overall quality.

Largest connected component. In the considered Gold Standard 1,000 gene networks, all nodes were connected. Moreover, in real biological data sets, it is often acknowledged that a GRN has a giant connected component [28] *e.g.* for robustness reasons. So being able to predict such a structure is a

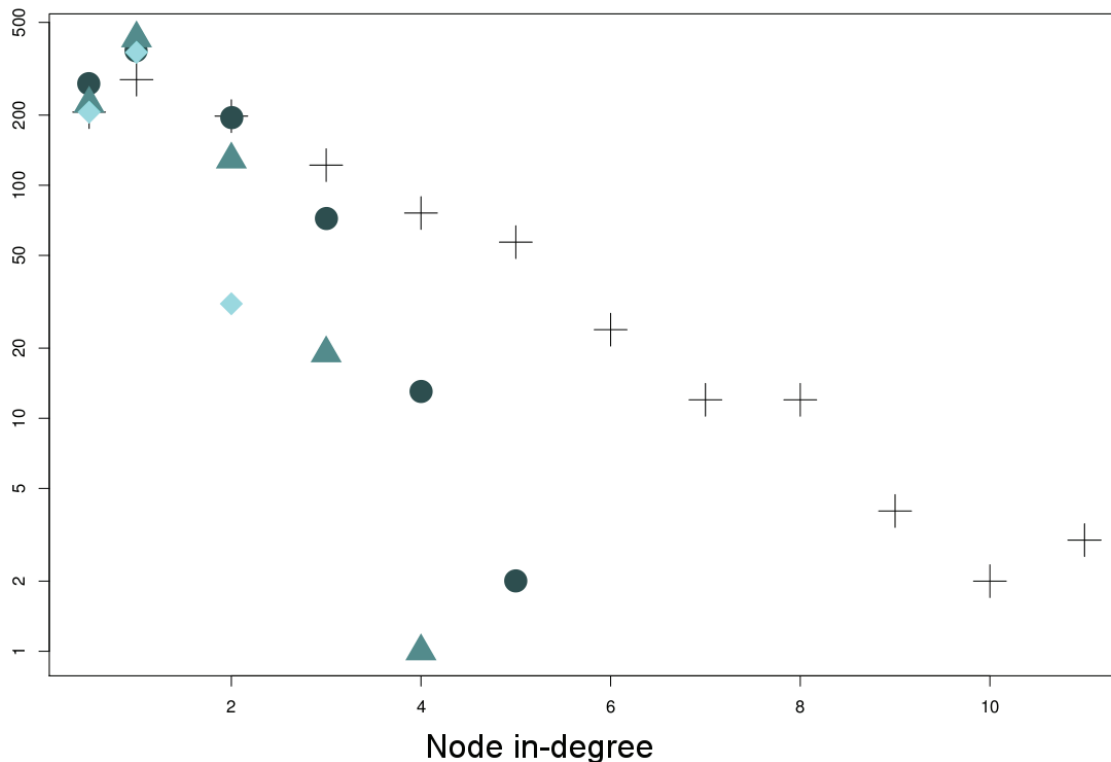**Figure 2. In-degree distribution of Network1-A999.** The distribution is plotted on the log scale on the *y*-axis since the in-degree distribution was assumed to be exponential in the true network (black crosses). Coloured symbols stand for the first 500 (light green diamond shape), 1,000 (middle green triangles) and 2,000 (dark green circles) edges inferred by the meta-analysis.
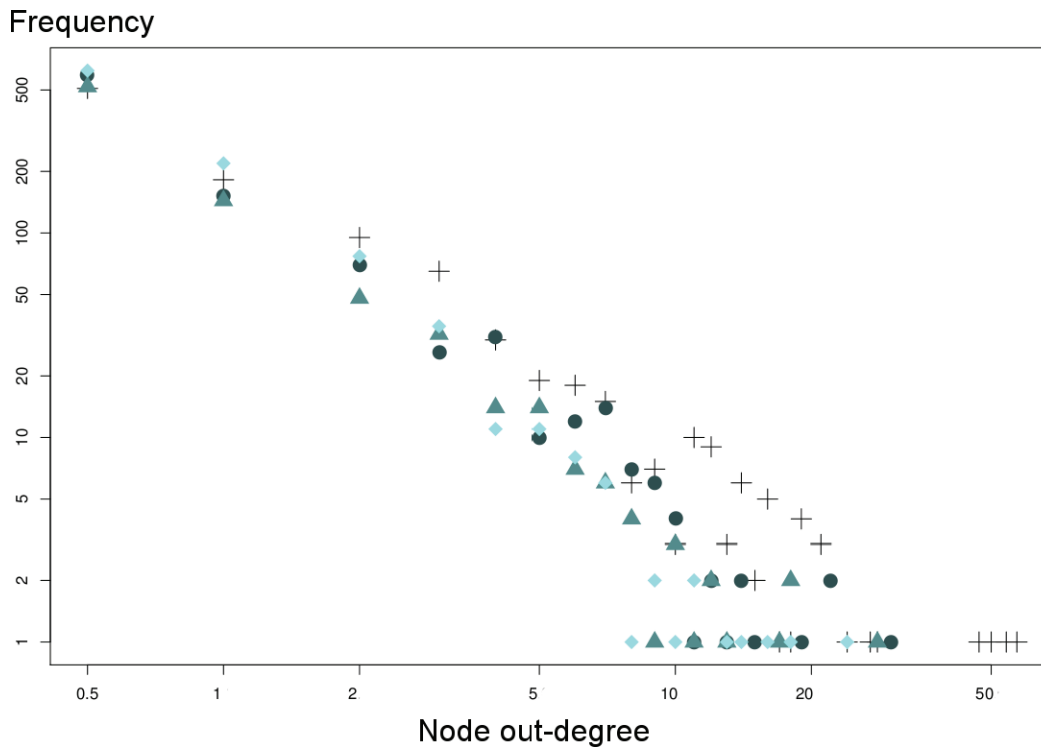doi:10.1371/journal.pone.0029165.g002

Frequency

Figure 3. Out-degree distribution of Network1-A999. The distribution is plotted on a log-log scale since it was expected to be a power-law distribution in the true network (black crosses). Coloured symbols stand for the first 500 (light green diamond shape), 1,000 (middle green triangles) and 2,000 (dark green circles) edges inferred by the meta-analysis. Points having '0' out-degree were transformed to 0.5.
doi:10.1371/journal.pone.0029165.g003

positive point for an inference method, even if all interactions are not simultaneously active [29]. The previous analysis on in-/out-degree also suggested to look at the size of the largest connected component when the number of considered edges increased. Figure 5 shows how this size evolves with the number of considered edges for the 15 different networks of the challenge. Clearly, three trends appear in our results, depending on sample size. In the 999 individuals case, the largest connected component captured almost all nodes when more than $1,500-2,000$ edges were considered, whatever the true graph connectivity. In the 100 individuals case, no large connected component appeared, even when considering 3,000 edges (with very low precision levels near 10%). For 300 individuals, it was a middle-of-the-road case. A large connected component appeared reasonably quickly with additional edges, but the behaviour changed with the true network connectivity.

Up to this point, we only presented global measures on the networks. In the next paragraph, we present results that show that prediction accuracy may also be influenced by local factors such as the type of mutation that occurred, either in the promoter region or in the coding sequence of the gene.

Edge inference accuracy depends on mutations that impact gene activity. We analysed the quality of inferred gene regulations depending on the type of mutation that occurred for the source (regulator) gene and the target (regulated) gene. We inferred the type of mutation of a gene: we labelled the gene '*cis*' if the mutation is in its promoter region (hence the mutation shows a *cis*-regulatory effect), and '*trans*' if it lies in its coding region. A *trans*-mutation modifies the sequence of the gene which, as a regulator, affects the expression of target genes in the GRN. This leads to a *trans*-regulatory effect. Some authors (*e.g.* [24]) call such regulation a '*cis-trans*' effect, but we used '*trans*' for simplicity.

For each gene, we tested the *cis*-regulatory effect of its marker using an analysis of variance, as described in the "Materials and Methods" section (Bayesian networks subsection). Genes not detected as *cis*-regulated were labelled '*trans*'. This gave a predicted number of *trans*-acting genes consistent with the announced frequency of 75% when the sample size was large enough to precisely infer this rate. When sample size was smaller, we underestimated *cis*-acting regulation frequency.

Figure 6 shows that '*cis*' → '*trans*' links were predicted more reliably than other types of relationships. This may be explained by the fact that the regulator of the target gene had a large variation due to the strong effect of its *cis* mutation, and that its regulatory effect was not obfuscated by a *cis*-regulation on the target gene. The '*cis*' → '*cis*' framework was the worst from the prediction accuracy point of view. It may correspond to strong correlations due to genetic linkage but not to direct causal regulations.

Complementarity of the inference methods combined in the meta-analysis. The meta-analysis took as input the inferred networks of three different methods: the Bayesian networks (BN), the Lasso regression, and the Dantzig selector-based regression. These methods ranked the edges differently and this was what allowed the meta-analysis to perform well.

Figure 7 displays a Venn diagram that presents specificity and overlaps between the sets of the first 1,000 edges predicted by the BN, Lasso, and Dantzig approaches, respectively. Similar figures were obtained with the first 500 or 2,000 edges instead (data not shown). It appeared that the edges simultaneously predicted by all three approaches were very reliable: 90% of them were correct. So were edges shared by the BN and Dantzig approaches. Edges predicted by just one method were less precise (less than 50% precision), and pure Lasso predictions were even poorer.
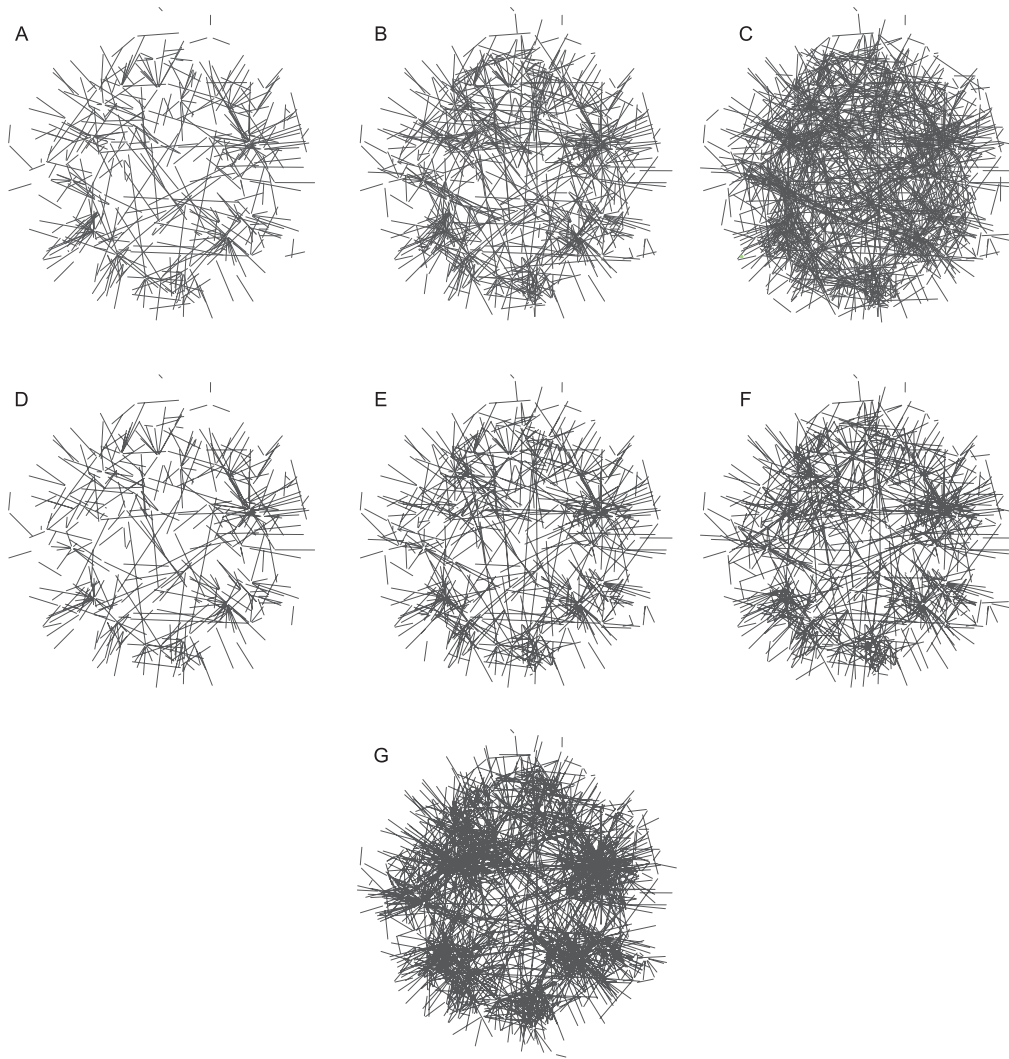
Figure 4. Network1-A999 visualisation. (A) to (C) are networks inferred by the meta-analysis using the first 500 (A), 1,000 (B) and 2,000 (C) edges. (D) to (F) represent the same predicted networks showing only correctly inferred edges. (G) is the true network. For clarity, vertices have been removed.
doi:10.1371/journal.pone.0029165.g004

The most interesting observation was that there was a clear complementarity between the three considered approaches. They shared a core prediction set, but each of them provided a specific contribution to the meta-analysis output.

Computing times. Table 2 presents computing times for the different approaches. These computation times were averaged over the five different networks in each of the three different sub-challenges. Computation times were very similar for the different networks, although the number of edges in the network to infer varied from less than 2,000 to more than 5,000. The CPU-times for the BN and Lasso methods had a nearly linear dependency upon sample size and should scale-up easily to larger data-sets. This seemed less obvious for the Dantzig selector approach but this was mostly because this recent method has been directly and bluntly implemented using a general linear programming solver. The use of a dedicated algorithm such as DASSO [30] would likely lead to an approach that scales as smoothly as the Lasso approach.

The meta-analysis is almost instantaneous as it only needs to parse edges lists for BN, Dantzig and Lasso to produce its own network and edge scores. However, it can not be run independently of other methods.

Discussion

We have proposed a GRN reconstruction method that relies on a meta-analysis of the output of three different reconstruction methods (namely BN, Lasso and Dantzig). As best performers of the DREAM5 Challenge 3A, we have shown that the presented methodology can adequately deal with large size gene network inference in a systems genetics (or genetical genomics) framework, *i.e.* when both marker data, that reflects mutations occurring in a segregating population, and gene expression data are available.

As expected, network reconstruction clearly improves when sample size increases. This is a decisive argument for planning genetical genomics experiments with enough individuals in the segregating population. Our results suggest that a sample of size 300 is at least needed to infer a first list of 500 reliable edges (at a precision level of nearly 65%) for a 1,000 gene network using the meta-analysis approach.
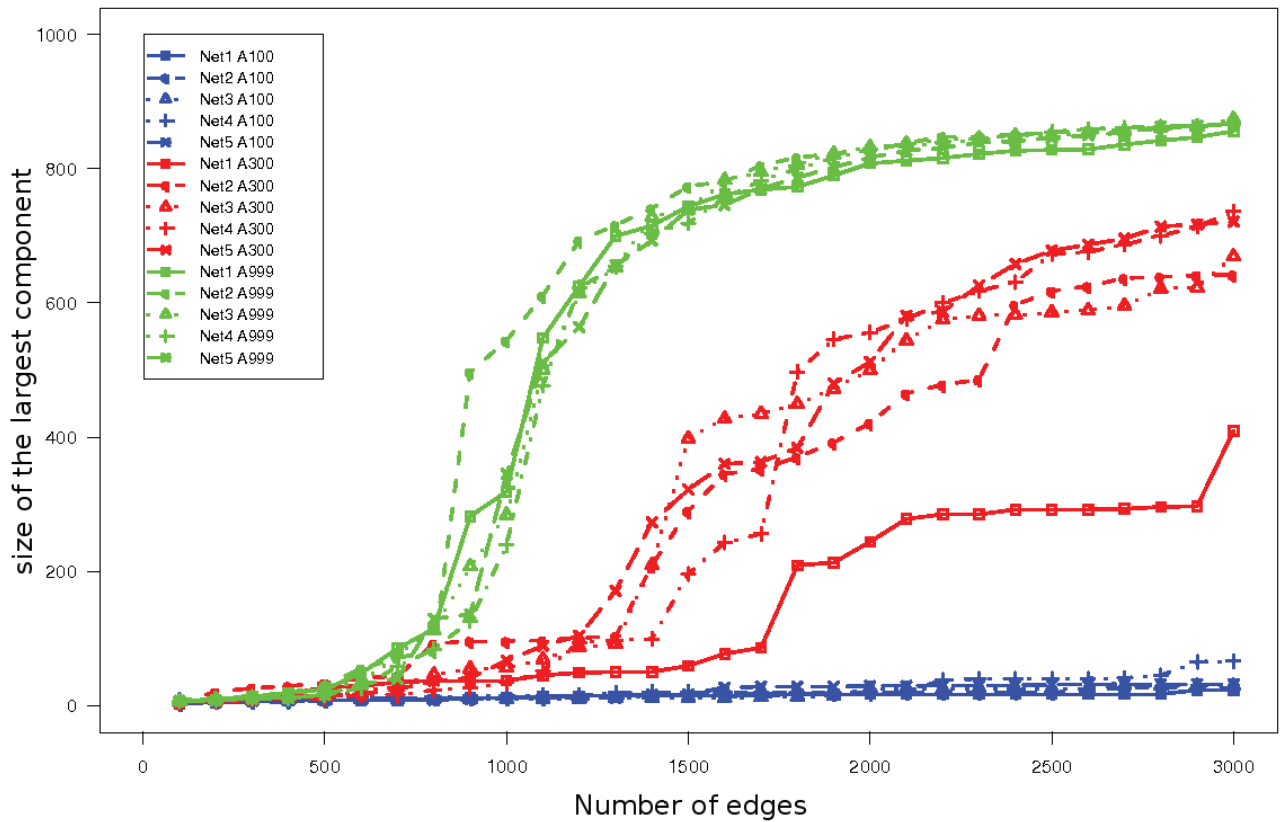
Figure 5. Size of the largest connected component inferred by the meta-analysis for the 15 **DREAM5 Challenge 3A networks vs. number of edges.** Colours encode sample sizes: blue for 100 individuals, red for 300 and green for 999. Line style and symbols on curves represent networks: solid line squares for Networks '1', short dashed line with circles for Networks '2', dotted line with triangles for Networks '3', alternate dashed and dotted line with plus for Networks '4' and long dashed line with crosses for Networks '5'.
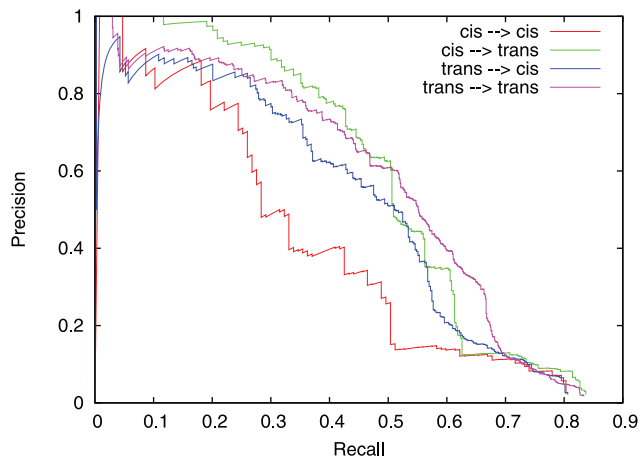doi:10.1371/journal.pone.0029165.g005

Figure 6. Analysis of precision/recall of the meta-analysis approach for DREAM5 Challenge 3A with 999 **individuals/ Network1 data.** Predicted gene regulations are classified into four groups depending on the label of the regulator and the target gene. A gene is labelled 'cis' if its marker has a cis-regulated effect on its expression level. Otherwise, gene is labelled 'trans'. An edge between two 'cis' labelled genes is classified 'cis → cis', between two 'trans' labelled genes: 'trans → trans' and so on.
doi:10.1371/journal.pone.0029165.g006

These good results could only be achieved thanks to the integration of three different complementary statistical inference techniques. This is certainly a key explanation for the results obtained. First of all, these predictions were produced by two different classes of methods, each capable of exploiting specific different features.

• structure learning of a Bayesian network: in this probabilistic framework, a directed acyclic graph is used to represent probabilistic relationships between discrete variables. The directed acyclic graph structure restricts the class of predicted GRN to structures that do not contain feedback loops. However, the use of directed graphs allows for predicting causal relationships between variables, as expected in the DREAM5 challenge.

Discrete Bayesian networks are also inherently limited by the usual encoding of probabilistic relationships between causes (parents) and effects using a conditional probability table for each node in the network. Such a table includes a number of parameters that grows exponentially with the number of causes. Since sample size is limited, only a limited number of parameters can be reliably estimated, and the approach is therefore inherently limited to graphs where every variable is explained only through a limited number of parents. For the Bayesian Information Criterion (BIC) a maximum number of 5 or 6 parents, depending on the choice of 3 or 4 classes for expression level variables, could be predicted with a sample size of 300 [31]. The BDeu score that we used for the structure learning is known to allow more parents [32], however we never attained the hard constraint of 9 parents that we imposed in our algorithm. This restriction was necessary
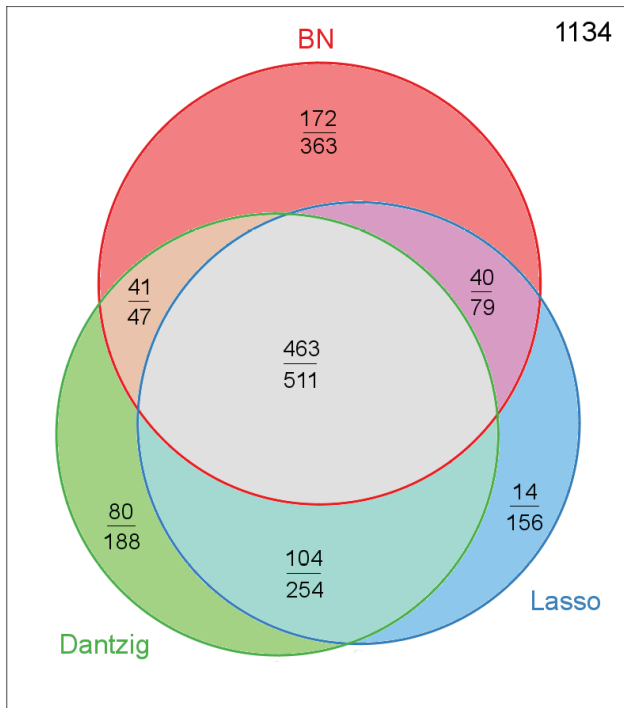
Figure 7. Venn diagram between the three sets made up of the first 1,000 edges inferred from one of the three approaches: BN (red circle), Lasso (blue circle) and Dantzig (green circle). Within each region of the diagram, the number of correctly inferred edges (over the bar) and the total number of edges (under the bar) are given. 1,134 (top right) is the number of missing edges for the union of the three approaches.
doi:10.1371/journal.pone.0029165.g007

for computational efficiency, as learning Bayesian network structure is NP-hard [33].

The positive part of this flexible encoding of probabilities distributions is that it enables the capture of non-linear relationships between variables, an expected behaviour of true biological samples.

• penalised linear regressions (Lasso and Dantzig): as a mirror to Bayesian networks, these approaches infer undirected graphs, with no causal relationships, but the predicted structures may contain cycles. They are restricted to linear relationships between variables, but this restriction keeps the number of parameters small. The number of neighbours of a variable is not *a priori* limited, and predicting hubs is possible. Ultimately, these models

Table 2. Computation times for the different approaches (per network).

Method	DREAM5 sub-challenge		
	A100	A300	A999
BN	20′	70′	180′
Lasso	5′	12′	30′
Dantzig	300′	1300′	6600′
Meta	less than a couple of seconds		

CPU times are given for a 2,96 Ghz Intel(TM) processor with 4 GB memory installed.
doi:10.1371/journal.pone.0029165.t002

are efficient in the sense that the associated inference algorithms are polynomial time algorithms [30,34].

Unexpectedly, despite the relationships between the two penalised linear regression methods [35], which should provide close estimates in a sparse setting, the Venn diagram in Figure 7 clearly shows that each method predicts different sets of edges showing complementarity even in their own class.

The idea of combining results from different methods has already been tested by the DREAM organisers themselves in a previous different DREAM challenge, in what they called "the community intelligence" [36]. With the best performers among the competitive teams, the DREAM organisers computed a a very simple and robust combined score based on rank sum. Their predictions outperformed individual teams when results of best performers were complementary and not optimal. We based our meta-analysis on a more sophisticated score that was accurate because our source methods had weighted edges with a probability-like score. Clearly, combining linear (Lasso/Dantzig) and non-linear (BN) methods allowed the meta-analysis we proposed to better detect causal relationships.

BN and penalised regressions also produced a very different total number of predicted edges. The number of predicted edge has a tremendous impact on DREAM challenge scores. BN predictions hardly reached a few thousand edges whilst Lasso and Dantzig approaches produced more than 100,000 edges each. To illustrate this, the area under the curve for true positive rate versus false positive rate in Figure 1 (top) was clearly smaller for BN predictions. Edge list of smaller length can be an explanation of poorer scores. The meta-analysis used the entire list of scores produced by BN, Lasso and Dantzig approaches. The edge ranking score (described in the "Material and Methods" section) we used gave better results than any of the individual approaches, except for the very first predictions (recall below 7% on Figure 1 bottom) ; in this latter situation, the Dantzig approach obtained slightly better precision (less than 1% improvement).

General features of the true network are ususally correctly recovered. For example, predicted networks have good in- and out-degree distributions and the expected construction of a big connected was quickly observed with only 1,000 or 2,000 predicted edges. In addition to individually ranking correct edges first, the meta-analysis is also able to retrieve global structural attributes of the network.

One obviously has to be careful about conclusions drawn from simulated data, as provided in the DREAM5 challenge. While experimental data on GRN slowly accumulates and expression measures become increasingly easy and inexpensive using RNA-Seq, to the best of our knowledge, neither sufficiently large experimental data sets that systematically combine gene expression and polymorphism measures, nor experimentally confirmed large GRN are available yet. In the area of genetical genomics data-sets, [37] exploited 160 RILs from a cross between two *Arabidopsis thaliana* accessions with 291 available markers and 24,065 gene transcript levels. Similarly, [38] gathered 45,000 gene expression levels, 194 micro-satellite markers for 60 F_2 mice and [39] recently analysed 110 RILs derived from a cross between two rice accessions with 1,655 markers and 16,372 expression traits. The three former examples were far from inferring a genome-scale GRN. These examples stress the gap between present research results obtained on real data sets that give only local regulatory relationships and simulation settings that indicate a potential for genome-scale GRN reconstruction on larger data sets. From our experience in analysing such data sets, several features can be quite different in real data sets and in simulated data sets, such as those proposed in the Challenge 3 of DREAM5.

One such feature is the unrealistic one marker per gene assumption: in practice, the total number of markers is unrelated to the number of genes and may be either quite low, or very high (see [40] for figures on plants and references therein with over a million SNPs for humans). A solution to the former case would be to infer pseudo-markers but still less comprehensive information would result from it. On the other hand, Next Generation Sequencing data sets are promising in that they would propose several markers per gene. Our modelling need be extended to use haplotypic markers instead of marker data to fully use the available multi-allelic information at different genotyped loci.

It should be pointed out that our prediction relies on probabilistic models, which are in no way related to the mechanistic ODE-based model used for generating the data set. In essence, none of our models is therefore using the "true model", which is the usual case when handling real data sets.

A potentially more challenging question lies in the number of genes in the network. As we have just pointed out, the analysis of a large number of genes requires large sample sizes, at higher costs. When dealing with GRN with thousands of genes and only a few hundreds of individuals in the population, the ultra-high dimension limit linking the sample size, the number of genes and the network sparsity is hit so that even sparse models can not be faithfully recovered anymore [41]. In the three formerly cited papers ([37–39]), if the number of parents/regressors associated to each gene was to exceed 4, the estimation would theoretically be impossible. Bootstrap techniques might help in providing sparse robust estimates in such settings [42–44]. A prior selection of relevant genes, using genes that are differentially expressed or selecting genes known to play a role in the biological process under study, could considerably improve GRN inference. The risk here is that if an important variable (*e.g.* integrative hub) is missing in the data set, confounding effects will likely lead to false positive edge predictions even when combining several methods into a powerful meta-analysis. The use of hidden variables, that could account for unmeasured gene expressions, has shown limited performances when the number of genes in the network is high (over a few tens of genes) ; interesting preliminary results can be found in [45,46]. There is a substantial need for methodological developments in this direction.

Following the added value of integrating several inference methods, a natural way to improve predictions would be to include additional inference methods which would complement the methods we used in the present study. Causality inference is probably the area where our current combination of inference tools could benefit from additional contributors. Indeed, the linear regression inferences essentially ignore causality, while Bayesian networks are able to predict causality when no Markov equivalence ambiguity appears. One should ideally be able to actively exploit the fact that the seed for causality from polymorphism to expression is known *a priori*. Before this, different existing inference techniques, such as kernel methods [47] and Random Forests [48], which have already been used in similar contexts [49,50], would be excellent candidates.

Materials and Methods

Notations and data simulation

The data sets provided by the DREAM5 Challenge 3 organisers are available at http://wiki.c2b2.columbia.edu/dream/index.php/D5c3.

Directed networks of $p = 1,000$ genes were generated according to a "modular scale-free topology" [18]. After the challenge, the organisers gave additional information on the generation process:

networks were simulated with a power law (scale free) out-degree distribution, but an exponential in-degree distribution. Moreover, simulated networks were modular. Fifteen such networks were generated and the distribution parameters were chosen so that the total number of edges range from 2,000 (Network '1' category) to 5,000 (Network '5' category). Each network was associated to a specific population size n of either $n = 100$ (sub-challenge A100), $n = 300$ (sub-challenge A300) and $n = 999$ (sub-challenge A999). In all cases, the sample size n was smaller than the total number of genes p and the $\frac{n}{p}$ ratio, which is important for estimation, varied from $\frac{1}{10}$ for sub-challenge A100 to $\frac{999}{1000}$ for sub-challenge A999.

For each network and each population size, genotypes for n RILs with 1,000 bi-allelic markers evenly distributed on 50 chromosomes were simulated using linkage information. Each RIL was an homozygous mosaic of paternal and maternal alleles. Parental alleles were different all along the genome. Each marker polymorphism was assumed to be associated with a single gene mutation located either in the promoter region (probability 25%) or in the coding region (probability 75%). A polymorphism in the promoter region of a gene affects its basal transcription rate, leading to a '*cis*-like' regulatory effect on the gene activity, while a polymorphism in the coding region affects the strength of the effect of the gene on its targets in the network, leading to a '*trans*-like' effect. The marker data for RIL $i \in \{1, \ldots, n\}$ and gene $j \in \{1, \ldots, p\}$ is denoted M_{ij} and has value 0 or 1. The genotype matrix M is hence a $n \times p$ matrix with 0/1 entries.

Gene expression levels were simulated at steady state of a dynamical system represented by a set of ODEs (see exact formula in [18] and details in [51]). These ODEs account, *via* different parametrisations, for different intensities in activation or repression effects, genetic variant influences and additional noise. The expression data matrix G consists in a $n \times p$ matrix.

Polymorphisms between RIL individuals define multifactorial perturbations. Each allelic combination defines a different parametrisation in the ODE model, with the same network skeleton. In addition to random term effects that represent technical or biological variability, this provokes changes in the observed gene expression patterns from one individual to another that in turn influence each other according to the causal network. The observed values G were those obtained at steady-state of the complex system. Figure 8 depicts such observed patterns for four individuals (1 to 4) of the population for Network1-A999 of the challenge.

The goal of the challenge was to reconstruct the network that gave birth to the observed measures as a list of edges sorted according to a relative "confidence" score. This score was only used for ranking edges.

We now present our strategy to reconstruct the networks namely the preprocessing of the data, the probabilistic graphical models that we implemented, their post-processing and the meta-analysis that was carried out to make the best out of the different modelling approaches.

Bayesian networks

Our first statistical modelling of the data relies on a directed graphical model known as Bayesian networks. Our model captures expression levels and genetic data in discrete variables, related through conditional probability tables capturing regulating and polymorphism effects, including possibly non-linear effects. The structure and parameters of the underlying graph were estimated using a score-based structure learning algorithm similarly to what was done in [13] in the context of pure expression data analysis.
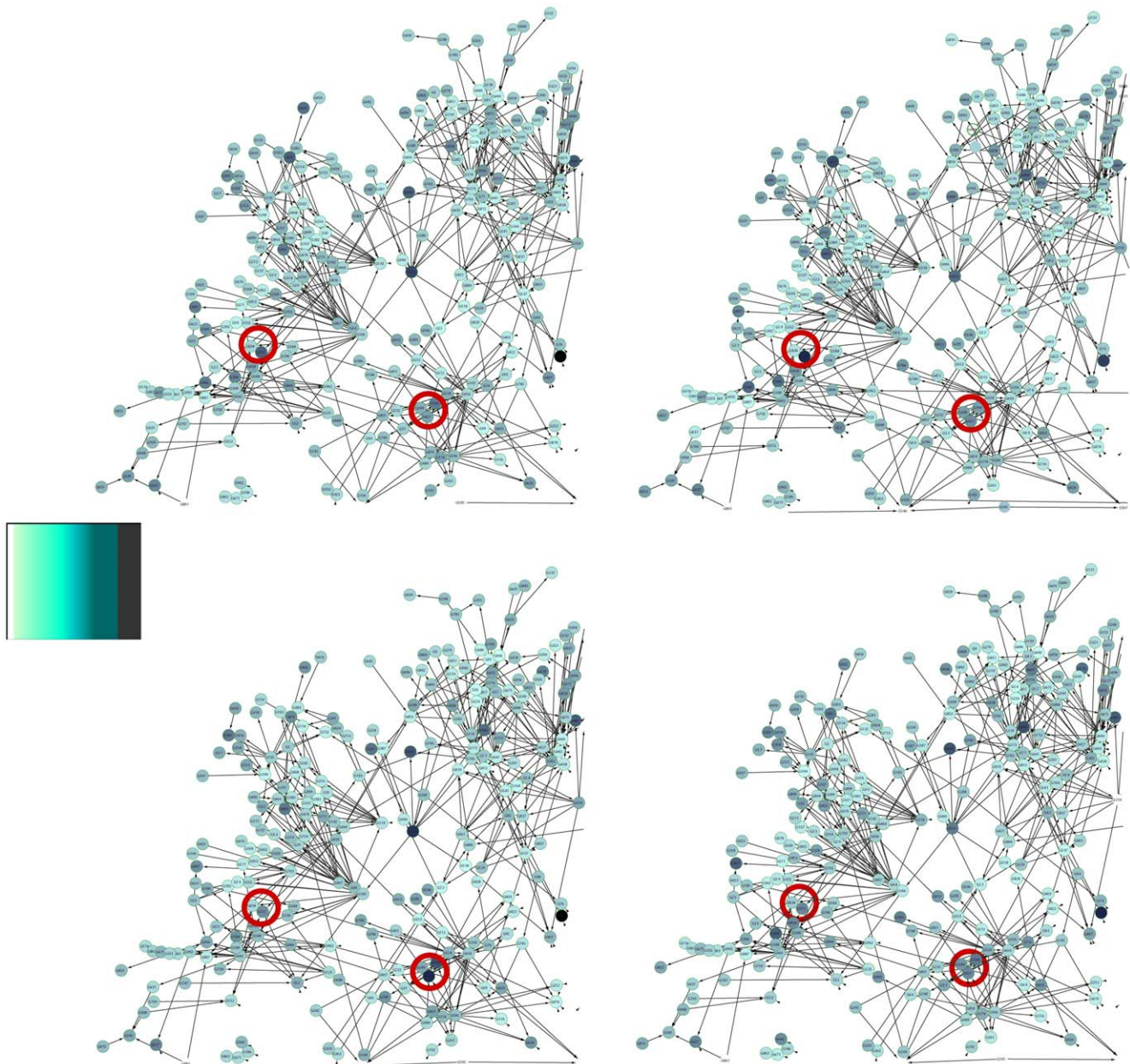
Figure 8. Graphical representation of expression data on a subpart of Network1-A999 for four individuals. Node colour represents simulated gene expression level (in green scale, light for small values and dark for high values) for individuals 1 (upper left), 2 (upper right), 3 (bottom left) and 4 (bottom right). Red circles highlight two spots in the network that vary due to different underlying marker polymorphisms.
doi:10.1371/journal.pone.0029165.g008

The precise score, discretisation policy, and algorithm used are described below.

Bayesian networks and structure learning with the Dirichlet score. A Bayesian network denoted by $B = (\mathcal{G}, P_{\mathcal{G}})$ is defined by a directed acyclic graph $\mathcal{G} = (V, E)$ with nodes representing p random discrete variables $V = \{V_1, \ldots, V_p\}$, linked by a set of directed edges E, and a set of conditional probability distributions $P_{\mathcal{G}} = \{P_1, \ldots, P_p\}$. The variables involved in each conditional probability table P_i are defined by the directed acyclic graph: $P_i = \mathbb{P}(V_i|Pa(V_i))$, where $Pa(V_i) = \{V_j \in V | (V_j, V_i) \in E\}$ is the set of parent nodes of V_i in $\mathcal{G}$. A Bayesian network B represents a joint probability distribution on V such that:

$$\mathbb{P}(V) = \prod_{i=1}^{p} \mathbb{P}(V_i|Pa(V_i)) \qquad (1)$$

Learning the structure of a Bayesian network consists in finding a directed acyclic graph $\mathcal{G}$ maximising $\mathbb{P}(\mathcal{G}|D)$ where D represents the observed data. We have:

$$\mathbb{P}(\mathcal{G}|D) = \frac{\mathbb{P}(D|\mathcal{G})\mathbb{P}(\mathcal{G})}{\mathbb{P}(D)} \propto \mathbb{P}(D|\mathcal{G})\mathbb{P}(\mathcal{G}) \qquad (2)$$

The first term $\mathbb{P}(D|\mathcal{G})$ of Equation 2, is called the marginal likelihood. The Bayesian Dirichlet score (BDeu, where 'eu' stands for Equivalent Uniform) gives the same score for Markov equivalent Bayesian networks and assumes a uniform prior on the conditional probability parameters. It is defined by the following expression:

$$BDeu(\mathcal{G}) = \mathbb{P}(D|\mathcal{G}) = \prod_{i=1}^{p} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(n_{ij} + \alpha_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(n_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})}$$

with n_{ijk}, the number of occurrences of the configuration $(V_i = k, Pa(V_i) = j)$ in the n samples, $n_{ij} = \sum_{k=1}^{r_i} n_{ijk}$, $\alpha_{ij} = \frac{\alpha}{q_i}$, and $\alpha_{ijk} = \frac{\alpha}{r_i q_i}$, where r_i is the domain size of variable V_i and q_i is the size of the Cartesian product of the i parent domains ($q_i = \Pi_{V_j \in Pa(V_i)} r_j$). The BDeu score requires a specific value for α, called the equivalent sample size, which, in practice, is often arbitrarily set to one. However, [52,53] established the sensitivity of the BDeu score with respect to this parameter: the connectivity of the inferred DAG increases with a growing α. [53] suggested a way to compute an optimal α value, making the assumption that α is smaller than the sample size. Following this idea, we defined a range of α values starting from its maximum value, set to the largest sample size (*i.e.* $\alpha = 10^3$), and decreasing it on a logarithmic scale. In our experiments, we varied α in the range $\{10^{-16}, 10^{-15}, \ldots, 10^3\}$ in order to get 20 networks, from a very sparse to a denser structure. This defined the α-grid for the Bayesian network approach. We defined the score of an edge using majority voting on these graphs (see below). Without any additional information, a uniform probability over all possible DAGs was assumed in Equation 2.

Bayesian network modelling and discretisation policy. The set of discrete random variables V was composed of one variable per gene-activity, denoted G_i, and one variable for each genetic marker, denoted M_i, for all $i \in \{1, \ldots, p\}$ with p the number of genes ($p = 1000$). Following challenge 3A assumption, each gene, with expression G_i, was associated with a single genetic marker M_i. Since we used discrete BN, we had to discretise G_i. As shown in [54], for the same score-based structure learning algorithm, the choice of a discretisation method can dramatically modify the quality of the inferred network. Instead of choosing a single discretisation method, we chose an adaptive method depending on the type of gene-activity distribution for each gene. Observing complex distributions in the data sets, we distinguished two types of distributions. If we detected a unimodal (normal-like) distribution, we used an adapted k-means algorithm to obtain a three-class discretisation, which also ensured a minimum class size (5% sample size) and a maximum size for extreme classes (30%). In the case of a multimodal distribution, we used the more general framework of Gaussian mixture models to find a maximum of four classes. Since the BDeu score depends on domain sizes, we tuned the parameters of our discretisation method to favour a four-class discretisation so that most of the G_i variables had the same domain size.

Structure learning and restricted search space. Learning Bayesian network is an NP-hard problem with a super-exponential search space of potential DAG structures [33] and even a greedy search heuristic method can be very time consuming when the number of variables p is large. In order to get reasonable computation times and also take into account biological knowledge, we reduced the search space by several assumptions.

A preliminary analysis of variance was used to predict *cis*-regulatory markers: detected positive markers (Bonferroni corrected p-value < 0.001) were those giving the most significant signal in

a 7 marker-width window, centred on the gene, to avoid false marker influence due to genetic linkage.

We used this *cis*-effect information to constrain structure search: since each *cis*-marker M_i had an effect on its associated gene activity G_i only, we constrained our model to use an $M_i \rightarrow G_i$ edge and forbade other edges outgoing from M_i. In the opposite case, when the marker M_i was not detected as *cis*-regulatory marker we only forbade the $M_i \nrightarrow G_i$ edge.

Following the approach of [55], for each gene expression G_i, we selected a list of candidate parents composed of genes G_j (resp. markers M_j) with a contribution to BDeu $\mathbb{P}(G_i|G_j)$ (resp. $\mathbb{P}(G_i|M_j)$) assuming a single parent G_j (resp. M_j) greater than $\mathbb{P}(G_i)$ assuming no parents. Moreover, due to the fact that markers in the same chromosome region had a tendency to be selected together because of linkage correlations, we chose the best marker in a 50 cMorgan sliding window. We did not try to learn edges between marker variables since it is useless for our purpose.

We used the structure learning algorithm 'greedy hill-climbing' of Banjo [56]. We started from an empty DAG and fixed a maximum number of parents to 9 to avoid overwhelming computational costs, in order to find the best DAG locally maximising Equation 2 for each value in the α-grid.

The directed edges from the resulting 20 DAGs learnt for the 20 different values of α were directly mapped onto genes to define a network relating the p genes: an edge from M_i to G_j in the learnt structure created an edge from G_i to G_j in the network. So, despite the fact that the underlying graphical model can only represent an acyclic directed structure, the final network may contain cycles. We computed the frequency of every directed edge in the inferred gene networks obtained by different values of the equivalent sample size α. This allowed us to perform a simple majority vote; directed edges were sorted based on their frequency, breaking ties by using average influence scores as defined in [57].

Structural equation modelling

This section first presents the structural equation model used to describe relationships among variables and the penalisation techniques that allowed for simultaneous parameter estimation and variable selection. We then explain how we implemented them in practice.

In the framework of Structural Equation Models (SEM), one response (or dependent) variable Y is assumed to depend upon m regressors X's with linear dependency in the parameters:

$$\mathbb{E}[Y|X] = f(X)$$
$$= \sum_{j=1}^{m} X_j \theta_j \quad (3)$$

Equation 3 is linear in parameters θ that are unknown and need to be estimated. Explanatory variables X can be quantitative or qualitative.

Having observed X and Y for a sample of size n, the usual estimation procedure is the ordinary least square (OLS) method which minimises the residual sum of squares (RSS):

$$RSS(\theta) = \sum_{i=1}^{n} (y_i - \sum_{j=1}^{m} x_{ij} \theta_j)^2$$
$$= \sum_{i=1}^{n} {}^{\mathrm{T}}(Y - X\theta)(Y - X\theta) \quad (4)$$
$$= ||Y - X\theta||_{\ell_2}^2$$

where y_i and x_{ij} are the observed values of Y and X_j for the ith individual.

Differentiating Equation (4) with respect to θ leads to the unique least square estimate

$$\hat{\theta}^{rss} = (^TXX)^{-1}\,^TXY \qquad (5)$$

On data sets where $n < m$, as in our case, X may be not full rank. ^TXX is then singular and the estimate in Equation (5) can be replaced by: $\hat{\theta} = X^\dagger Y$ where $X^\dagger$ is the Moore-Penrose pseudo inverse of X. Beyond intensive computation, another difficulty arises when eigenvalues of X close to 0 cannot be estimated precisely enough because of numerical instability, inducing large uncertainty about $X^\dagger$ and consequently $\hat{\theta}$. Ultimately, the unbiased estimate $\hat{\theta}$ is not worth the effort because of the induced variance in coefficient estimates or predictions.

Since our goal was to obtain an interpretable (*i.e.* reasonable number of significant explanatory variables) and stable (*i.e.* small changes in data should have low impact on analysis results) model that leads to as accurate as possible predictions, we allowed some bias-variance trade-off using regularization. One approach to regularization is to introduce a constraint on regression coefficients. For example ridge regression minimises the RSS in Equation 4 imposing that $||\theta||_{\ell_2} \leq t$ for some $t > 0$. The smaller t, the greater the level of shrinkage of regression coefficients. Conversely, larger values of t allow more complex models that are penalised if they do not bring enough gain into the RSS. The model selection problems therefore come down to choosing appropriate values for t. However, ridge regression does not select variables: every regression coefficient is shrunk but not set to 0 so the model is not really simpler. Direct variable selection procedures, such as "best subset" tackle the issue of a huge number of regressors included in the model but as a discrete process they can be subject to a high variance in the produced estimates.

Among the many possible techniques to achieve variable selection in a stable model, we chose to focus on the Lasso [25] and on the Dantzig Selector [26]. Other penalization techniques (for example see [43,58,59]) are known to be more suitable for high dimensional data that have inherent inner collinearity, but need additional parameter tuning. Our goal was first to focus on simple, efficient but powerful techniques in order to assess their merits on this problem.

Lasso penalised regression. The Lasso is very similar to ridge regression. It also minimises the RSS but allows for deviation up to a penalty term controlled by a constraint on the ℓ_1 norm of parameters θ (instead of the ℓ_2-norm for ridge regression). The Lasso automatically selects variables and continuously shrinks their associated regression coefficients. Depending on the penalization strength, it enforces an increasing number of parameters to be 0. Lasso estimates are defined as follows:

$$\hat{\theta}^{Lasso} = \arg\min_\theta ||Y - X\theta||_{\ell_2}, \text{ subject to } ||\theta||_{\ell_1} \leq t \qquad (6)$$

or equivalently (Lagrangian transform):

$$\hat{\theta}^{Lasso} = \arg\min_\theta ||Y - X\theta||_{\ell_2}^2 + \lambda ||\theta||_{\ell_1} \qquad (7)$$

While Equation (6) explicits the constraint on the parameters norm, Equation (7) introduces the penalty parameter λ: the larger λ, the greater the amount of shrinkage, and the simpler the selected model will be. More precisely, λ is an upper bound on the correlation between regressors not included in the model and the regression residual. Interpreting t of Equation (6) is also possible by considering $t_0 = ||\hat{\theta}^{rss}||_{\ell_1}$. Hence setting t to $t_0/2$ roughly shrinks active coefficients in the regression by 50% [44]. The Lasso solutions do not vary equally upon input scales. Standardisation of the inputs settles this problem. For the Lasso (and for the Dantzig selector below), we therefore standardised the input regressors.

Solving Equation (7) is a quadratic programming problem but efficient algorithms exist for computing the entire solution path as λ varies. We used the the Least Angle Regression (LAR, [34]) algorithm available in the glmnet package version 1.4 [60] and implemented in R (version 2.11.0, http://www.r-project.org/).

In the challenge, confidence scores had to be assigned to inferred edges, so we did not use a model selection criterion but instead created a score reflecting the importance of the explanatory variable. This score was the frequency for this variable to be included in the model for different values of the penalization parameter. This could be done along the entire LAR solution path. We used a fixed grid of λ values. For comparability with the Dantzig selector and BN approaches, we used a grid of 20 evenly spaced values for λ, ranging from 0 (no penalization) to $\lambda_{\max}$, the smallest value of λ that prevents any regressor to be included in any regression.

Dantzig selector. The Dantzig selector [26] is a recent regression method which, as the Lasso approach, relies on the ℓ_1 norm of the parameters to capture model complexity. In its standard description, the Dantzig selector minimises the ℓ_1 norm of the parameters subject to constraints bounding the absolute value of the correlation between residuals and explanatory variables. Similarly to the definition of the Lasso given in Equations (6) and (7), the Dantzig Selector is:

$$\hat{\theta}^{Dantzig} = \arg\min_\theta \left\{ ||\theta||_{\ell_1} : ||^TX(Y - X\theta)||_{\ell_\infty} \leq \delta \right\} \qquad (8)$$

where δ is a bound on the correlation between the residual vector and each explanatory variable. With no bound ($\delta \to \infty$), the Dantzig selector estimates all coefficients to zero, because of the minimised ℓ_1 norm. With the strongest bound ($\delta = 0$), Dantzig enforces a zero correlation between residuals and explanatory variables, a condition which is also satisfied by ordinary least square estimates (as it is equivalent to enforcing a zero derivative of the squared error term minimised in OLS regression). Equation (8) can be written in its dual form:

$$\hat{\theta}^{Dantzig} = \arg\min_\theta \left\{ ||^TX(Y - X\theta)||_{\ell_\infty} : ||\theta||_{\ell_1} \leq t \right\} \qquad (9)$$

This writing is similar to the Lasso of Equation (6), replacing the RSS by the maximum varying component of its gradient.

As initially shown in [26], the Dantzig selector is able to produce an accurate estimate in the $n \ll p$ context with a bounded error term, provided that the model is actually sparse. The Dantzig selector also has the property that it reduces regression to linear programming, a polynomial optimisation problem [61] for which efficient dedicated solvers exist. Recently, [35] showed that the Lasso and the Dantzig selector share similar properties: the Lasso

estimate automatically satisfies Dantzig correlation constraints, and similar error boundaries can be obtained in both cases (although with larger constant terms for the Lasso). For this reason, the Dantzig selector tends to be considered as extremely similar to the Lasso. However [44] noticed that coefficient regularization paths are quite smooth along Lasso solution while they can become irregular with the Dantzig selector.

To solve each regression problem, we generated a linear program (LP) as described in [26]. The generated LP was far from optimised. Beyond the p variables for vector θ, it included n variables for the residual vector and p variables to encode boundaries on correlations. $2p$ boundaries were used to effectively limit correlations and an extra set of n linear equalities encoding residual definition. The size of this encoding depended on p and n but could easily depend just on p (by symbolic precomputation of the scalar product between residuals and explanatory variables).

By setting all parameters θ to 0 while minimising δ, it is simple to compute the minimum value of δ such that all regression coefficients are set to 0 (denoted as $\delta_{\max}$). We then solved the Dantzig selector problem using the GPL linear programming solver glpk for 20 evenly spaced values of δ in $[0, \delta_{\max}]$ (for comparability reasons with the BN and Lasso approaches), providing a set of 19 non-trivial estimates for the parameters in θ.

Application of structural equation models to systems genetics data. We now show how we used penalised regressions to infer a GRN from the DREAM5 Challenge 3A data sets.

We regressed each gene expression level G_i for $i \in \{1 \ldots p\}$ using as regressors every other gene expression level and every gene marker. This gene-by-gene approach ignores correlations and therefore corresponds to the minimisation of a specific penalised pseudo-likelihood [62]. Its main advantage is to reduce the whole penalised likelihood minimisation to p univariate penalised linear regressions.

Let G denotes the $n \times p$ observed matrix of the gene expression levels and M the $n \times p$ matrix of marker genotypes; the linear regression model for gene i is:

$$G_i = G\beta_i + M\alpha_i + \varepsilon_i \qquad (10)$$

$$= \sum_{\substack{g=1 \\ g \neq i}}^{p} \beta_{ig} G_g + \sum_{g=1}^{p} \alpha_{ig} M_g + \varepsilon_i$$

where β_i is the p-vector of linear effects of other expression levels on G_i (forcing $\beta_{ii} = 0$ to avoid trivial self-regression), α_i is the p-vector of linear effects of markers on G_i and ε_i is the Gaussian residual error term.

To make the link with previous notations in this section, Y now iteratively becomes one of the G_i variables, X becomes the $n \times 2p$ matrix (G, M) and the regression coefficients θ now become (β, α). The network is then encoded in non-zero entries of estimated matrices $\beta := [\beta_1, \cdots, \beta_p]$ and $\alpha := [\alpha_1, \cdots, \alpha_p]$. The only consistency condition is that $\beta_{ii} = 0$ for all i.

From the estimated β and α matrices, the gene-to-gene network was inferred. More precisely, when $\alpha_{ij} \neq 0$ for some i and $j \in \{1 \ldots p\}$, we inferred edge $i \rightarrow j$ in the gene network and assigned it a count of 1. If both α_{ij} and α_{ji} are equal to zero, then the β matrix was explored. If $\beta_{ij} \neq 0$ or $\beta_{ji} \neq 0$, we inferred both edges $i \rightarrow j$ and $j \rightarrow i$ and assigned them a count of $1/2$. Finally we computed for each edge, the count mean in the chosen λ-grid for Lasso or on the δ-grid for Dantzig. This means that we put a high confidence level in directed edges inferred from marker to a gene expression level and that we inferred edges between gene

expression data by symmetrising and halving their strength. This choice is somewhat arbitrary and can certainly be improved, as we commented in the discussion.

Meta-analysis: integrating several network inference methods

We used a Fisher's Inverse Chi-Square meta-test [63] to combine the BN, Lasso and Dantzig predicted networks. This meta-test was initially introduced to combine the test values obtained from independent experiments. It consists of summing the opposite of the logarithm of the corresponding p-values.

In the output data for the DREAM challenge, we considered the "reliability" parameter as 1 minus p-value since it is a measure of uncertainty in the $[0; 1]$ range. We denoted $\mathcal{M} = \{BN, Lasso, Dantzig\}$ and $\left(r_{ij}^m\right)$ the edge reliability parameters associated to the method $m \in \mathcal{M}$. We then computed the sum

$$S_{ij} = \sum_{m \in \mathcal{M}} \log\left(1 - r_{ij}^m\right)$$

The meta-analysis picks up edges from the different approaches and computes a consensus ranking scheme that depends on individual scores of the methods and agreement between them.

Finally the meta-analysis edge reliability parameters are defined as $r_{ij} = 1 - \exp(S_{ij})$ and were used to produce a ranked list a edges for each inferred network. Since the organisers limited the edge list length that could be submitted to 100,000 among the 999,000 possible edges in each network (no self loops were considered), we arbitrarily cut the list according to the ranking when necessary. In practice, we never predicted more than 107,000 edges.

Accuracy assessment: scoring methods

Once submitted to the DREAM5 challenge organisers, edge lists were compared both to (i) Gold Standard files, namely the correct list of edges used in simulated models and to (ii) the pool of all edges that were submitted by other participating teams.

The Gold Standard comparison allows to assess prediction accuracy based on two measures, namely the "area under the curve" (AUC) score for the Receiver Operating Curve (ROC *i.e.* true positive versus false positive rates) and the precision versus recall (PR) curve. The second comparison evaluates predictions on the basis of their intrinsic merit and on their ability to bring in specific predictions compared with the pool of all predicted edges.

Let TP, FP, FN and TN denote respectively the true positives (correctly inferred edges), false positives (edges inferred by mistake), false negatives (missed edges) and true negatives (correctly non-predicted edges), then (i) False positive Rate ($= \frac{FP}{FP+TN}$, (ii) Precision $= \frac{TP}{TP+FP}$ and (iii) Recall = True positive rate $= \frac{TP}{TP+FN}$. Notice that the orientation of edges is significant in the comparison so that an edge $g_1 \rightarrow g_2$ is not considered as correct if the true edge is $g_2 \rightarrow g_1$.

The second comparison assesses the prediction of every method against the pool of all the predictions submitted by competing teams. It was used to produce p-values that reflect how well each method performed in comparison to others. More precisely, the lower the p-value for a team prediction AUC, the higher the probability that it could not be reached by a random network built by picking up edges (at the same rank) from the pool of all submitted networks. The p-values for all 15 different networks were then log-transformed and summed in absolute value. The

higher the resulting score, the better the method performed over the challenge. A detailed description of the scoring scheme for the DREAM5 challenges can be found in [27].

Acknowledgments

Author Contributions

Conceived and designed the experiments: MV JV DA CCA TS BM SdG. Performed the experiments: MV JV DA NRA TS BM SdG. Analyzed the data: MV JV DA TS BM SdG. Contributed reagents/materials/analysis tools: MV JV DA NRA CCA TS BM SdG. Wrote the paper: MV JV DA CCA TS BM SdG.

References

1. Brazhnik P, de la Fuente A, Mendes P (2002) Gene networks: how to put the function in genomics. Trends in Biotechnology 20: 467–472.
2. Yvert G, Brem R, Whittle J, Akey J, Foss E, et al. (2003) Trans-acting regulatory variation in saccharomyces cerevisiae and the role of transcription factors. Nature genetics 35: 57–64.
3. Leclerc R (2008) Survival of the sparsest: robust gene networks are parsimonious. Molecular Systems Biology 4.
4. Marbach D, Mattiussi C, Floreano D (2009) Replaying the evolutionary tape: biomimetic reverse engineering of gene networks. Annals of the New York Academy of Sciences 1158: 234–245.
5. Siegal M, Promislow D, Bergman A (2007) Functional and evolutionary inference in gene networks: does topology matter? Genetica 129: 83–103.
6. Hecht J, Stricker S, Wiecha U, Stiege A, Panopoulou G, et al. (2008) Evolution of a core gene network for skeletogenesis in chordates. PLoS Genetics 4.
7. Kauffman S (1969) Homeostasis and differentiation in random genetic control networks. Nature 224: 177–178.
8. Thomas R (1973) Boolean formalization of genetic control circuits. Journal of Theoretical Biology 42: 563–585.
9. Barabási AL, Oltvai Z (2004) Network biology: understanding the cells functional organization. Nature Reviews Genetics 5: 101–113.
10. Dhaeseleer1 P, Liang S, Somogyi R (2000) Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics.
11. Gardner T, di Bernardo D, Lorenz D, Collins J (2003) Inferring genetic networks and identifying compound mode of action via expression profiling. Science 301: 102–105.
12. Xiong M, Li J, Fang X (2004) Identification of genetic networks. Genetics 166: 1037–1062.
13. Friedman N, Linial M, Nachman I, Peer D (2000) Using Bayesian networks to analyse expression data. Journal of Computational Biology 7: 601–620.
14. Bansal M, di Bernardo D (2007) Inference of gene networks from temporal gene expression profiles. IET Systems Biology 1: 306–312.
15. Rau A, Jaffrezic F, Fouley JL, Doerge R (2010) An empirical bayesian method for estimating biological networks from temporal microarray data. Statistical Applications in Genetics and Molecular Biology 9: 9.
16. Lèbre S, Becq J, Devaux F, Stumpf M, Lelandais G (2010) Statistical inference of the time-varying structure of gene-regulation networks. BMC Systems Biology 4.
17. Werhli A, Husmeier D (2008) Gene regulatory network reconstruction by Bayesian integration of prior knowledge and/or different experimental conditions. Journal of Bioinformatics and Computational Biology 6: 543–572.
18. de la Fuente A, Stolovitzky G D5c3 - Dream initiative. Available: http://wiki.c2b2.columbia.edu/dream/index.php/D5c3. Organizers: Columbia university and IBM.
19. Fisher R (1935) The Design of Experiments. EdinburghLondon: Oliver and Boyd.
20. Jansen R, Nap N (2001) Genetical genomics : the added value from segregation. Trends in Genetics 17: 388–391.
21. Jansen R (2003) Studying complex biological systems using multifactorial perturbation. Nature Reviews in Genetics 4: 145–151.
22. Aten J, Fuller T, Lusis A, Horvath S (2008) Using genetic markers to orient the edges in quantitative trait networks: the NEO software. BMC Bioinformatics 2.
23. Zhu J, Wiener M, Zhang C, Fridman A, Minch E, et al. (2007) Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. PLoS Computational Biology 3: e69.
24. Liu B, de la Fuente A, Hoeschele I (2008) Gene network inference via structural equation modeling in genetical genomics experiments. Genetics 178: 1763–1776.
25. Tibshirani R (1996) Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Methodological) 58: 267–288.
26. Candès E, Tao T (2007) The Dantzig selector: Statistical estimation when p is much larger than n. Annals of Statistics 35: 2313–2351.
27. Stolovitzky G, Prill R, Califano A (2009) Lessons from the DREAM2 challenges. In: Stolovitzky G, Kahlem P, Califano A, eds. Annals of the New York Academy of Sciences volume 1158: 159–195.
28. Ciliberti S, Martin O, Wagner A (2007) Robustness can evolve gradually in complex regulatory gene networks with varying topology. PLoS Computational Biology 3: e15.
29. Chaves M, Sontag E, Albert R () Structure and timescale analysis in genetic regulatory networks. In: Proc. of the IEEE Conf. Decision and Control. San Diego, 2006. pp 2358–2363.
30. James G, Radchenko P, Jinchi L (2009) DASSO : connections between the Dantzig selector and lasso. Journal of the Royal Statistical Society B 71: 127–142.
31. de Campos C, Zeng Z, Ji Q (2009) Structure learning of Bayesian networks using constraints. In: Proc. of ICML '09. pp 113–120.
32. de Campos C, Ji Q (2010) Properties of Bayesian Dirichlet scores to learn Bayesian network structures. In: Fox M, Poole D, eds. Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. Atlanta, Georgia, USA: AAAI Press. pp 431–436.
33. Chickering D, Heckerman D, Meek C (2004) Large-sample learning of Bayesian networks is NPhard. The Journal of Machine Learning Research 5: 1287–1330.
34. Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Annals of Statistics 32: 407–499.
35. Bickel P, Ritov Y, Tsybakov A (2009) Simultaneous analysis of lasso and Dantzig selector. Annals of statistics 37: 1705–1732.
36. Prill R, Marbach D, Saez-Rodriguez J, Sorger P, Alexopoulos L, et al. (2010) Towards a rigorous assessment of systems biology models: The DREAM3 challenges. PLOS ONE 5: e9202.
37. Keurentjes J, Fu J, Terpstra I, Garcia J, van den Ackerveken G, et al. (2007) Regulatory network construction in Arabidopsis by using genome-wide gene expression quantitative trait loci. Proceedings of the National Academy of Sciences 104: 1708–1713.
38. Chen HLM, Flowers J, Yandell B, Stapleton D, Mata C, et al. (2006) Combined expression trait correlations and expression quantitative trait locus mapping. PLoS Genetics 2: e6.
39. Wang J, Yu H, Xie W, Xing Y, Yu S, et al. (2010) A global analysis of qtls for expression variations in rice shoots at the early seedling stage. The Plant Journal 63: 1063–1074.
40. Weir B (2007) Impact of dense genetic marker maps on plant population genetic studies. Euphytica 154: 355–364.
41. Wainwright M (2009) Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. IEEE Transactions on Information Theory 55: 5728–5741.
42. Friedman J (2001) Greedy function approximation: a gradient boosting machine. Annals of Statistics.
43. Bach F (2008) Bolasso: model consistent lasso estimation through the bootstrap. In: Cohen W, McCallum A, Roweis S, eds. Proceedings of the Twenty-fifth International Conference on Machine Learning (ICML). Helsinki, Finland, volume 307 of ACM International Conference Proceeding Series, pp. 25–32.
44. Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Series in Statistics. Springer, second edition.
45. Elidan G, Nachman I, Friedman N (2007) "ideal parent" structure learning for continuous variable Bayesian networks. Journal of Machine Learning Research 8: 1799–1833.
46. Mooij J, Stegle O, Janzing D, Zhang K, Schölkopf B (2010) Probabilistic latent variable models for distinguishing between cause and effect. In: Lafferty J, Williams C, Shawe-Taylor J, Zemel R, Culotta A, eds. Advances in Neural Information Processing Systems 23: 1687–1695.
47. Schölkopf B, Tsuda K, Vert JP, eds (2004) Kernel Methods in Computational Biology. MIT PRess.
48. Breiman L (2001) Random forests. Machine Learning 45: 5–32.
49. Lippert C, Stegle O, Ghahramani Z, Borgwardt KM (2009) A kernel method for unsupervised structured network inference. In: Dyk DV, Welling M, eds. Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics MIT Press. pp 358–365.
50. Huynh-Thu V, Irrthum A, Wehenkel L, Geurts P (2010) Inferring regulatory networks from expression data using tree-based methods. PLoS ONE 5: e12776.
51. Pinna A, Soranzo N, Hoeschele I, de la Fuente A (2011) Simulating systems genetics data with SysGenSIM. Bioinformatics 27: 2459–2462.
52. Silander T, Kontkanen P, Myllymäki P (2007) On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. In: Proc. of UAI-07. Vancouver, Canada, pp360–367.
53. Steck H (2008) Learning the Bayesian network structure: Dirichlet prior vs data. In: UAI. pp 511–518.

54. Yong L, Lili L, Xi B, Hua C, Wei J, et al. (2010) Comparative study of discretization methods of microarray data for inferring transcriptional regulatory networks. BMC Bioinformatics 11: 520.

55. Friedman N, Nachman I, Peér D (1999) Learning bayesian network structure from massive datasets: The "sparse candidate" algorithm. In: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence. StockholmSweden, . pp 206–215.

56. Hartemink A (2005) Reverse engineering gene regulatory networks. Nature Biotechnology 23: 554–555.

57. Yu J, Smith V, Wang P, Hartemink A, Jarvis E (2002) Using Bayesian network inference algorithms to recover molecular genetic regulatory networks. In:Proceedings of the Third International Conference on Systems Biology (ICSB02). Stockholm, Sweden.

58. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society Series B (Methodological) 67: 301–320.

59. Zou H (2006) The adaptive lasso and its oracle properties. Journal of the American Statistical Association 101: 1418–1429.

60. Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models. Journal of Statistical Software 33.

61. Karmarkar N (1984) A new polynomial-time algorithm for linear programming. Combinatorica 4: 373–395.

62. Rocha G, Zhao P, Yu B (2008) A path following algorithm for sparse pseudo-likelihood inverse covariance estimation (SPLICE). Technical Report 769, Statistics Department, UC Berkeley.

63. Hedges L, Olkin I (1985) Statistical methods for meta-analysis. Academic Press.