Genomics 97 (2011) 7–18

Contents lists available at ScienceDirect

Genomics

journal homepage: www.elsevier.com/locate/ygeno

A comprehensive assessment of methods for de-novo reverse-engineering of genome-scale regulatory networks

Varun Narendra [a], Nikita I. Lytkin [a], Constantin F. Aliferis [a,b,c], Alexander Statnikov [a,d,*]

[a] Center for Health Informatics and Bioinformatics, New York University School of Medicine, New York, NY 10016, USA
[b] Department of Pathology, New York University School of Medicine, New York, NY 10016, USA
[c] Department of Biostatistics, Vanderbilt University, Nashville, TN 37232, USA
[d] Department of Medicine, New York University School of Medicine, New York, NY 10016, USA

ARTICLE INFO

Article history:
Received 31 August 2010
Accepted 7 October 2010
Available online 14 October 2010

Keywords:
Regulatory network de-novo
reverse-engineering
Computational methods
Evaluation
Gene expression microarray analysis

ABSTRACT

De-novo reverse-engineering of genome-scale regulatory networks is an increasingly important objective for biological and translational research. While many methods have been recently developed for this task, their absolute and relative performance remains poorly understood. The present study conducts a rigorous performance assessment of 32 computational methods/variants for de-novo reverse-engineering of genome-scale regulatory networks by benchmarking these methods in 15 high-quality datasets and gold-standards of experimentally verified mechanistic knowledge. The results of this study show that some methods need to be substantially improved upon, while others should be used routinely. Our results also demonstrate that several univariate methods provide a "gatekeeper" performance threshold that should be applied when method developers assess the performance of their novel multivariate algorithms. Finally, the results of this study can be used to show practical utility and to establish guidelines for everyday use of reverse-engineering algorithms, aiming towards creation of automated data-analysis protocols and software systems.

1. Introduction

The cell is a dynamic system of molecules that interact and regulate each other. Discovering these regulatory interactions is essential to expanding our understanding of normal and pathologic cellular physiology, and can lead to the development of drugs that manipulate cellular pathways to fight disease. A global model of gene regulation will also be essential for the design of synthetic genomes with targeted properties, such as the production of biofuels and medically relevant molecules [11]. There exist many databases that encapsulate biological pathways (e.g., KEGG and BioCarta); however, these databases are often inaccurate and incomplete and do not correspond to the studied biological system and experimental conditions [1,29,42,45]. Therefore, there is a strong need for the reverse-engineering of genome-scale regulatory networks using de-novo methods.

There is no doubt that data from targeted knockout/overexpression and/or longitudinal experiments provide the richest information about gene interactions that can be used by de-novo reverse-engineering methods. Unfortunately, such data is not currently abundant enough to enable discovery of regulatory networks, whereas there are thousands of available observational datasets from non-longitudinal case–control or case–series studies [7,39]. In addition, obtaining data from targeted knockout/overexpression experiments can be more expensive, unethical and/or infeasible for many biological systems and conditions. Thus, current methods are forced to utilize non-longitudinal case–control or case–series genome-scale data to reverse-engineer regulatory networks.

Over the last decade, many methods have been developed to reverse-engineer regulatory networks from observational data. However, their absolute and relative performance remains poorly understood [34]. Typically, a study that introduces a novel method performs only a small-scale evaluation using one or two datasets [21], and without comprehensive benchmarking against the best-performing techniques [38]. Such studies can show that the novel method is promising, but cannot demonstrate its empirical superiority and utility in general. Similarly, the past international competitions for reverse-engineering of regulatory networks have not provided a definitive answer as to what the best performing techniques are for genome-scale non-longitudinal observational data. The only competition that used real gene expression data for the inference of genome-scale network was DREAM2 [46]. However, since this competition involved a single dataset to which many methods had been applied, the results may be overfitted and thus may not generalize to other datasets [18].

The present study, for the first time, conducts a rigorous performance assessment of methods for reverse-engineering of

* Corresponding author. Center for Health Informatics and Bioinformatics, New York University Langone Medical Center, 227 E30th Street, 7th Floor, Office #736, New York, NY 10016, USA. Fax: +1 212 263 5995.
E-mail address: Alexander.Statnikov@med.nyu.edu (A. Statnikov).

0888-7543/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ygeno.2010.10.003

Table 1
Combined PPV and NPV metric (Euclidean distance from the optimal algorithm with PPV = 1 and NPV = 1) for 30 methods/variants over 15 datasets. Methods denoted "Full Graph" and "Empty Graph" output the fully connected and empty regulatory networks, respectively. Details about other methods and their parameters can be found in Table 8 and Table S3 in the Online Supplement. Cells with lighter highlighting correspond to less accurate algorithms; cells with darker highlighting correspond to more accurate algorithms.

Method		REGED	GNW(A)	GNW(B)	GNW(C)	GNW(D)	ECOLI(A)	ECOLI(B)	ECOLI(C)	ECOLI(D)	YEAST(A)	YEAST(B)	YEAST(C)	YEAST(D)	YEAST(E)	YEAST(F)
Aracne	$\alpha = 10^{-7}$	0.350	0.796	0.725	0.840	0.864	0.851	0.862	0.826	0.858	0.969	0.970	0.972	0.958	0.962	0.963
	$\alpha = 0.05$	0.826	0.802	0.739	0.841	0.868	0.851	0.862	0.826	0.858	0.969	0.970	0.972	0.958	0.962	0.963
Relevance Networks 1	$\alpha = 10^{-7}$	0.995	0.953	0.888	0.965	0.942	0.985	0.985	0.980	0.975	0.980	0.982	0.983	0.973	0.977	0.980
	$\alpha = 0.05$	0.997	0.981	0.950	0.985	0.979	0.986	0.986	0.981	0.981	0.980	0.982	0.983	0.973	0.977	0.980
Relevance Networks 2		0.994	0.937	0.903	0.954	0.948	0.984	0.984	0.979	0.968	0.979	0.981	0.983	0.973	0.977	0.979
SA - CLR	$\alpha = 0.05$	0.976	0.944	0.880	0.949	0.933	0.960	0.963	0.956	0.953	0.978	0.980	0.982	0.972	0.976	0.978
	FDR = 0.05	0.718	0.858	0.762	0.873	0.868	0.899	0.908	0.893	0.882	0.970	0.971	0.974	0.962	0.965	0.968
CLR	*Normal MI estimator;* $\alpha = 0.05$	0.963	0.928	0.850	0.933	0.913	0.951	0.957	0.947	0.947	0.979	0.981	0.982	0.973	0.977	0.978
	Normal MI estimator; FDR = 0.05	0.693	0.846	0.737	0.855	0.849	0.887	0.901	0.879	0.888	0.972	0.972	0.974	0.965	0.969	0.970
	Stouffer MI estimator; $\alpha = 0.05$	0.975	0.934	0.858	0.939	0.920	0.959	0.963	0.955	0.953	0.979	0.981	0.982	0.973	0.977	0.978
	Stouffer MI estimator; FDR = 0.05	0.736	0.858	0.751	0.866	0.859	0.911	0.922	0.907	0.905	0.974	0.975	0.976	0.967	0.971	0.972
LGL - Bach	*max - k = 1, w/o symmetry*	0.185	0.528	0.665	0.720	0.788	0.552	0.577	0.495	0.611	0.949	0.956	0.950	0.936	0.944	0.935
	max - k = 2, w/o symmetry	0.141	0.571	0.655	0.724	0.565	0.429	0.400	0.356	0.568	0.939	0.941	0.940	0.930	0.942	0.938
	max - k = 3, w/o symmetry	0.127	0.553	0.655	0.734	0.559	0.540	0.521	0.403	0.578	0.928	0.937	0.927	0.921	0.938	0.928
	max - k = 1, with symmetry	0.173	0.528	0.663	0.722	0.790	0.600	0.609	0.508	0.608	0.950	0.957	0.951	0.938	0.945	0.936
	max - k = 2, with symmetry	0.105	0.556	0.655	0.712	0.566	0.509	0.494	0.415	0.557	0.931	0.934	0.923	0.926	0.935	0.921
	max - k = 3, with symmetry	0.087	0.524	0.616	0.522	0.543	0.465	0.439	0.378	0.559	0.941	0.938	0.932	0.927	0.933	0.921
Hierarchical Clustering		0.996	0.944	0.850	0.950	0.914	0.960	0.964	0.956	0.956	0.979	0.981	0.982	0.973	0.976	0.979
Graphical Lasso		0.801	0.393	0.384	0.608	0.686	0.805	0.840	0.786	0.301	0.970	0.973	0.973	0.964	0.969	0.966
GeneNet	$\alpha = 0.05$	0.975	0.974	0.938	0.982	0.972	0.965	0.971	0.961	0.961	0.971	0.972	0.973	0.963	0.967	0.969
	FDR = 0.05	0.805	0.970	0.943	0.977	0.969	0.895	0.912	0.887	0.891	0.960	0.961	0.961	0.951	0.956	0.956
qp - graphs	*q = 1*	0.996	0.979	0.946	0.984	0.977	0.986	0.986	0.981	0.981	0.980	0.982	0.983	0.973	0.977	0.980
	q = 2	0.996	0.980	0.949	0.985	0.978	0.986	0.986	0.981	0.981	0.980	0.982	0.983	0.973	0.978	0.980
	q = 3	0.996	0.981	0.949	0.985	0.979	0.986	0.986	0.981	0.981	0.980	0.984	0.985	0.973	0.978	0.981
	q = 20	0.995	0.981	0.950	0.985	0.979	0.986	0.986	0.981	0.981	0.980	0.982	0.983	0.973	0.977	0.980
	q = 200	0.996	0.979	0.949	0.983	0.977	0.986	0.986	0.981	0.981	0.980	0.982	0.983	0.973	0.977	0.980
Fisher	$\alpha = 0.05$	0.996	0.975	0.935	0.980	0.972	0.984	0.985	0.979	0.978	0.980	0.982	0.983	0.973	0.977	0.980
	FDR = 0.05	0.996	0.973	0.932	0.979	0.971	0.984	0.985	0.979	0.978	0.980	0.982	0.984	0.973	0.977	0.980
Full Graph		0.998	0.981	0.952	0.985	0.979	0.986	0.986	0.981	0.981	0.980	0.982	0.983	0.973	0.977	0.980
Empty Graph		0.998	0.981	0.952	0.985	0.979	0.986	0.986	0.981	0.981	0.980	0.982	0.983	0.973	0.977	0.980

genome-scale regulatory networks by benchmarking state-of-the-art methods (from bioinformatics/systems biology and quantitative disciplines such as computer science and biostatistics) in multiple high-quality datasets and gold-standards of experimentally verified mechanistic knowledge. The results of this study can be used to show practical utility and to establish guidelines for everyday use of network reverse-engineering algorithms, with ancillary benefits providing guidance about "best of breed" inference engines suitable for automated data-analysis protocols and software systems.

2. Results

This work assessed the accuracy of 32 state-of-the-art network reverse-engineering methods/variants in 15 genome-scale real and simulated datasets/gold-standards. Since reverse-engineering methods are used in a variety of contexts, a single metric cannot be used to assess their accuracy. In order to capture the broad applicability of reverse-engineering algorithms, four benchmarks were conducted in this study, and each of them used a different metric to evaluate accuracy of reverse-engineering (details about metrics are provided in Materials and methods section). In each benchmark, methods were ranked based on their accuracy, and the top-five scoring methods were considered "best of breed". Methods that were winners in at least one of the four benchmarks should be used routinely by biologists and bioinformaticians for reverse-engineering purposes, while other methods should be substantially improved upon.

2.1. Benchmark #1: Which methods have the best combined positive predictive value (PPV) and negative predictive value (NPV)?

Implementations of LGL-Bach, regardless of parameters, constituted all of the top-five performing techniques (Tables 1 and 5). This method output few regulatory interactions relative to the size of the gold-standard. However, a larger percentage of these were true-positive interactions than for any other algorithm. Moreover, for most datasets >98–99% of the regulatory interactions not predicted by LGL-Bach did not exist in the gold-standard. Such a relatively accurate list of putative interactions can be fruitful for biologists because it limits the case of experimentally validating the false-positive interactions of a reverse-engineering method. Of note, Graphical Lasso performed the best on the GNW(A), GNW(B), and ECOLI(D) datasets. However, it performed poorly on all other datasets, and therefore ranks only seventh among all methods.

2.2. Benchmark #2: Which methods have the best combined sensitivity and specificity?

The methods that produced the best combined sensitivity and specificity were Relevance Networks 2, CLR (Stouffer MI estimator; $\alpha = 0.05$), Fisher (FDR = 0.05), SA-CLR ($\alpha = 0.05$), and CLR (Normal MI estimator; $\alpha = 0.05$) (Tables 2 and 5). These methods discovered more true regulatory interactions than LGL-Bach did. However, this came at the expense of a larger proportion of false-positive interactions. Biologists with limited resources may prefer results from methods such as LGL-Bach that are less complete (i.e., with smaller sensitivity), but more accurate (i.e., with larger PPV). Of note, Relevance Networks 1 produced the best performing results on three of the four GNW datasets and ECOLI(D). However, its poor performance on the other ECOLI datasets and REGED lowered its overall ranking to seventh. LGL-Bach and Aracne had the best performance among all methods on REGED, but performed poorly on all other datasets.

2.3. Benchmark #3: Which methods have the best area under the ROC (AUROC) curve?

The area under the ROC curve was measured for the 12 methods/variants that produce scores for graph edges (Table 3), and it provides a threshold-independent metric of the classification power[1] of each method. In order from first to fifth place, the best performing algorithms were *qp*-graphs ($q = 200$), CLR (Normal MI estimator), CLR (Stouffer MI estimator), *qp*-graphs ($q = 20$), and MI 2 (Table 5). Notably, the Fisher method produced top-5 AUROC scores over all REGED, GNW, and ECOLI datasets, but performed statistically indistinguishably from random on YEAST datasets. It is important to note that *qp*-graphs performed very well with respect to the threshold-independent AUROC metric, but very poorly in terms of the combined sensitivity and specificity. This discrepancy accentuates the difficulty in choosing an optimal threshold for this method as discussed below.

2.4. Benchmark #4: Which methods have the best area under the precision–recall (AUPR) curve?

The area under the precision–recall curve was also measured for all 12 score-based methods/variants (Table 4). Methods that perform well according to this metric produce a list of putative interactions that strike a balance between recall (or sensitivity) and precision (or PPV). CLR (Stouffer MI estimator) and CLR (Normal MI estimator) were the best performing methods, occupying first and second places, respectively. *qp*-graphs ($q = 200$) ranked third, SA-CLR ranked fourth, and MI 2 ranked fifth (Table 5). Notably, MI 2 turned out to be among the top performing methods because of its performance in REGED, GNW, and ECOLI datasets; its performance in YEAST datasets was statistically indistinguishable from random.

2.5. Some methods often outperform other techniques, while others are consistent underperformers

Operationally we define a method to be an underperformer if it did not score in the top-5 methods/variants for at least one of the four performance metrics. According to our study, the underperforming methods are Aracne, Relevance Networks 1, Hierarchical Clustering, Graphical Lasso, GeneNet, and MI 1. This implies that other state-of-the-art algorithms can produce better results across a wide range of gold-standards/datasets and performance metrics. Hence, the underperforming algorithms should be revisited and substantially improved upon.

Since there is no single performance metric that fully captures the power of a method in all conceivable contexts of application, all algorithms that scored well with respect to at least one metric should be used in the context in which they performed best. Our analysis shows that CLR is a top performer for three metrics; *qp*-graphs, SA-CLR, and MI 2 are top performers for two metrics; while LGL-Bach, Relevance Networks 2, and Fisher are top performers with respect to one metric.

2.6. Univariate methods[2] provide a "gatekeeper" performance threshold and should be used when method developers assess performance of their novel algorithms

In the last several years there has been an emergence of mathematically and computationally complex novel methods for reverse-engineering of regulatory networks [34,45,46]. We believe

[1] In this context, classification power refers to the ability to correctly classify each pair of genes as having a direct regulatory interaction, or not.

[2] A univariate method refers to a method that only tests for pairwise association between a target gene and a single gene.

Table 2
Combined sensitivity and specificity metric (Euclidean distance from the optimal algorithm with sensitivity = 1 and specificity = 1) for 30 methods/variants over 15 datasets. Methods denoted as "Full Graph" and "Empty Graph" output the fully connected and empty regulatory networks, respectively. Details about other methods and their parameters can be found in Table 8 and Table S3 in the Online Supplement. Cells with lighter highlighting correspond to less accurate algorithms; cells with darker highlighting correspond to more accurate algorithms.

Method		REGED	GNW(A)	GNW(B)	GNW(C)	GNW(D)	ECOLI(A)	ECOLI(B)	ECOLI(C)	ECOLI(D)	YEAST(A)	YEAST(B)	YEAST(C)	YEAST(D)	YEAST(E)	YEAST(F)
Aracne	$\alpha = 10^{-7}$	0.019	0.936	0.913	0.867	0.891	0.938	0.943	0.944	0.941	0.991	0.990	0.990	0.991	0.990	0.989
	$\alpha = 0.05$	0.021	0.935	0.911	0.865	0.889	0.938	0.943	0.944	0.941	0.991	0.990	0.990	0.991	0.990	0.989
Relevance Networks 1	$\alpha = 10^{-7}$	0.432	0.556	0.612	0.425	0.450	0.959	0.956	0.962	0.603	0.836	0.837	0.837	0.835	0.836	0.836
	$\alpha = 0.05$	0.887	0.970	0.966	0.977	0.972	1.000	1.000	1.000	0.994	1.000	1.000	1.000	1.000	1.000	1.000
Relevance Networks 2		0.403	0.612	0.575	0.466	0.467	0.802	0.807	0.804	0.612	0.750	0.753	0.753	0.754	0.755	0.755
SA - CLR	$\alpha = 0.05$	0.093	0.639	0.712	0.557	0.578	0.732	0.741	0.754	0.738	0.869	0.867	0.865	0.875	0.873	0.868
	FDR = 0.05	0.074	0.802	0.867	0.768	0.791	0.853	0.867	0.867	0.866	0.986	0.986	0.986	0.987	0.986	0.986
CLR	*Normal MI estimator; $\alpha = 0.05$*	0.060	0.609	0.673	0.525	0.565	0.762	0.779	0.782	0.768	0.903	0.903	0.900	0.905	0.904	0.899
	Normal MI estimator; FDR = 0.05	0.049	0.724	0.814	0.677	0.752	0.858	0.873	0.876	0.871	0.981	0.979	0.978	0.982	0.982	0.980
	Stouffer MI estimator; $\alpha = 0.05$	0.088	0.595	0.657	0.511	0.539	0.744	0.757	0.763	0.746	0.871	0.872	0.868	0.875	0.874	0.867
	Stouffer MI estimator; FDR = 0.05	0.043	0.709	0.799	0.657	0.729	0.848	0.863	0.867	0.858	0.973	0.971	0.969	0.974	0.974	0.972
LGL - Bach	*max - k = 1, w/o symmetry*	0.018	0.987	0.958	0.948	0.949	0.980	0.984	0.983	0.981	0.999	0.999	0.999	0.999	0.999	0.999
	max - k = 2, w/o symmetry	0.022	0.984	0.964	0.950	0.967	0.988	0.990	0.991	0.984	0.998	0.998	0.998	0.999	0.999	0.998
	max - k = 3, w/o symmetry	0.030	0.984	0.958	0.945	0.968	0.985	0.987	0.986	0.982	0.998	0.998	0.998	0.998	0.999	0.998
	max - k = 1, with symmetry	0.021	0.987	0.958	0.948	0.950	0.978	0.981	0.980	0.981	0.999	0.999	0.999	0.999	0.999	0.999
	max - k = 2, with symmetry	0.028	0.984	0.963	0.951	0.967	0.986	0.987	0.987	0.986	0.998	0.998	0.998	0.999	0.999	0.998
	max - k = 3, with symmetry	0.048	0.984	0.962	0.965	0.968	0.988	0.991	0.991	0.984	0.999	0.999	0.998	0.999	0.999	0.998
Hierarchical Clustering		0.594	0.889	0.917	0.814	0.802	0.826	0.834	0.833	0.856	0.871	0.868	0.867	0.876	0.869	0.865
Graphical Lasso		0.050	0.963	0.963	0.909	0.919	0.910	0.924	0.923	0.966	0.993	0.993	0.992	0.994	0.993	0.992
GeneNet	$\alpha = 0.05$	0.121	0.848	0.896	0.890	0.894	0.753	0.783	0.801	0.793	0.838	0.831	0.818	0.844	0.839	0.827
	FDR = 0.05	0.260	0.957	0.994	0.987	0.996	0.874	0.890	0.903	0.895	0.948	0.946	0.941	0.953	0.951	0.946
qp - graphs	*q = 1*	0.611	0.779	0.799	0.883	0.853	0.997	0.997	0.997	0.966	0.997	0.997	0.997	0.997	0.997	0.997
	q = 2	0.619	0.867	0.889	0.949	0.914	1.000	1.000	1.000	0.990	1.000	1.000	1.000	1.000	1.000	1.000
	q = 3	0.621	0.908	0.923	0.967	0.941	1.000	1.000	1.000	0.996	1.000	1.000	1.000	1.000	1.000	1.000
	q = 20	0.447	0.968	0.968	0.986	0.971	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	q = 200	0.543	0.700	0.694	0.676	0.677	0.999	1.000	1.000	0.991	0.999	0.999	0.999	0.998	0.999	0.999
Fisher	$\alpha = 0.05$	0.573	0.598	0.601	0.625	0.613	0.783	0.791	0.787	0.691	0.807	0.810	0.808	0.808	0.810	0.809
	FDR = 0.05	0.558	0.568	0.575	0.591	0.579	0.773	0.782	0.778	0.674	0.800	0.803	0.802	0.801	0.803	0.802
Full Graph		1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Empty Graph		1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table 3
Area under ROC curve (AUROC) for 12 methods/variants over 15 datasets. Cells with bold values correspond to AUROC estimates that are statistically different from random (AUROC = 0.5) according to the method of DeLong et al. [17]. Details about methods and their parameters can be found in Table 8 and Table S3 in the Online Supplement. Cells with lighter highlighting correspond to less accurate algorithms; cells with darker highlighting correspond to more accurate algorithms.

Method		REGED	GNW(A)	GNW(B)	GNW(C)	GNW(D)	ECOLI(A)	ECOLI(B)	ECOLI(C)	ECOLI(D)	YEAST(A)	YEAST(B)	YEAST(C)	YEAST(D)	YEAST(E)	YEAST(F)
SA-CLR		**0.996**	**0.641**	**0.605**	**0.724**	**0.721**	**0.637**	**0.632**	**0.604**	**0.619**	**0.509**	**0.509**	**0.509**	0.499	0.501	0.505
CLR	*Normal MI estimator*	**0.996**	**0.695**	**0.674**	**0.762**	**0.755**	**0.640**	**0.631**	**0.620**	**0.624**	**0.510**	**0.509**	**0.509**	0.503	0.505	**0.508**
	Stouffer MI estimator	**0.996**	**0.696**	**0.670**	**0.762**	**0.759**	**0.636**	**0.629**	**0.611**	**0.616**	**0.510**	**0.509**	**0.509**	0.503	0.505	**0.509**
GeneNet		**0.971**	**0.550**	**0.545**	**0.524**	**0.530**	**0.599**	**0.578**	**0.571**	**0.570**	**0.542**	**0.546**	**0.552**	**0.539**	**0.541**	**0.550**
qp-graphs	*q = 1*	**0.901**	**0.651**	**0.649**	**0.694**	**0.681**	**0.614**	**0.595**	**0.593**	**0.615**	**0.509**	0.507	**0.511**	**0.505**	0.505	**0.511**
	q = 2	**0.947**	**0.658**	**0.654**	**0.701**	**0.681**	**0.622**	**0.604**	**0.597**	**0.618**	**0.511**	**0.510**	**0.514**	**0.507**	**0.508**	**0.514**
	q = 3	**0.968**	**0.661**	**0.655**	**0.705**	**0.686**	**0.626**	**0.608**	**0.601**	**0.621**	**0.512**	**0.511**	**0.516**	**0.508**	**0.509**	**0.516**
	q = 20	**0.998**	**0.668**	**0.638**	**0.696**	**0.673**	**0.635**	**0.623**	**0.611**	**0.623**	**0.530**	**0.536**	**0.545**	**0.521**	**0.527**	**0.540**
	q = 200	**0.999**	**0.586**	**0.529**	**0.586**	**0.557**	**0.648**	**0.641**	**0.635**	**0.629**	**0.549**	**0.556**	**0.559**	**0.540**	**0.547**	**0.555**
Fisher		**0.993**	**0.671**	**0.664**	**0.716**	**0.699**	**0.634**	**0.611**	**0.606**	**0.618**	0.496	0.492	0.494	0.494	0.492	0.495
MI 1		**0.992**	**0.683**	**0.671**	**0.767**	**0.756**	**0.644**	**0.630**	**0.620**	**0.624**	0.497	0.493	0.491	0.492	0.489	0.489
MI 2		**0.991**	**0.688**	**0.673**	**0.759**	**0.752**	**0.653**	**0.630**	**0.630**	**0.631**	0.502	0.499	0.499	0.495	0.495	0.497

Table 4
Area under precision–recall curve (AUPR) for 12 methods/variants over 15 datasets. Cells with bold values correspond to AUPR estimates that are statistically different from random according to the method of [43]. Details about methods and their parameters can be found in Table 8 and Table S3 in the Online Supplement. Cells with lighter highlighting correspond to less accurate algorithms; cells with darker highlighting correspond to more accurate algorithms.

Method		REGED	GNW(A)	GNW(B)	GNW(C)	GNW(D)	ECOLI(A)	ECOLI(B)	ECOLI(C)	ECOLI(D)	YEAST(A)	YEAST(B)	YEAST(C)	YEAST(D)	YEAST(E)	YEAST(F)
SA-CLR		**0.514**	**0.102**	**0.131**	**0.102**	**0.104**	**0.087**	**0.082**	**0.089**	**0.097**	**0.023**	**0.022**	**0.021**	**0.029**	**0.025**	**0.023**
CLR	*Normal MI estimator*	**0.551**	**0.156**	**0.180**	**0.161**	**0.141**	**0.087**	**0.081**	**0.091**	**0.096**	**0.022**	**0.021**	**0.020**	**0.028**	**0.025**	**0.023**
	Stouffer MI estimator	**0.544**	**0.157**	**0.180**	**0.161**	**0.142**	**0.086**	**0.079**	**0.090**	**0.096**	**0.023**	**0.021**	**0.020**	**0.029**	**0.025**	**0.023**
GeneNet		**0.595**	**0.023**	**0.057**	**0.017**	**0.024**	**0.077**	**0.070**	**0.075**	**0.069**	**0.026**	**0.025**	**0.024**	**0.033**	**0.029**	**0.028**
qp-graphs	*q = 1*	**0.012**	**0.038**	**0.090**	**0.034**	**0.044**	**0.022**	**0.021**	**0.028**	**0.032**	**0.022**	**0.020**	**0.018**	**0.028**	**0.024**	**0.022**
	q = 2	**0.022**	**0.043**	**0.096**	**0.038**	**0.046**	**0.025**	**0.022**	**0.030**	**0.034**	**0.022**	**0.020**	**0.019**	**0.029**	**0.025**	**0.023**
	q = 3	**0.036**	**0.047**	**0.101**	**0.042**	**0.051**	**0.026**	**0.023**	**0.032**	**0.036**	**0.022**	**0.020**	**0.019**	**0.029**	**0.025**	**0.023**
	q = 20	**0.446**	**0.084**	**0.127**	**0.072**	**0.072**	**0.041**	**0.035**	**0.045**	**0.059**	**0.025**	**0.023**	**0.022**	**0.032**	**0.028**	**0.027**
	q = 200	**0.917**	**0.039**	**0.063**	**0.029**	**0.033**	**0.082**	**0.073**	**0.078**	**0.093**	**0.028**	**0.027**	**0.026**	**0.035**	**0.032**	**0.030**
Fisher		**0.784**	**0.126**	**0.162**	**0.116**	**0.101**	**0.073**	**0.061**	**0.079**	**0.086**	0.020	0.018	0.016	0.026	0.022	0.020
MI 1		**0.769**	**0.140**	**0.171**	**0.084**	**0.135**	**0.073**	**0.064**	**0.077**	**0.078**	0.020	0.018	0.016	0.026	0.022	0.020
MI 2		**0.789**	**0.140**	**0.166**	**0.089**	**0.106**	**0.079**	**0.068**	**0.082**	**0.090**	0.020	0.018	0.016	0.026	0.022	0.020

that the cost of added complexity should be offset by an increased performance of the method. Hence, the simplest (univariate) methods should provide a "gatekeeper" performance threshold, above which all novel complex algorithms should perform.

With respect to the combined positive and negative predictive value metric, the added complexity of the winning LGL-Bach method is justified by its superior performance compared to the highest-ranking univariate method (CLR, only ninth place). Similarly, *qp*-graphs ($q = 200$) achieve a better AUROC than any univariate method, and should be used despite its increased complexity. On the other hand, the three top performing methods with respect to the combined sensitivity and specificity are all univariate methods. Similarly, the univariate CLR method performs optimally with respect to AUPR. Therefore, researchers interested in methods that currently produce the best results with respect to the above two metrics do not need to use computationally more expensive multivariate methods.

2.7. It is challenging to select an optimal threshold for a method that outputs scores for edges rather than a network graph

Recall that the score-based methods output scores for all possible edges in a graph. A regulatory network is then obtained by choosing a threshold and pruning all edges whose scores are below the threshold. Therefore, the quality of the produced network largely depends on the choice of a threshold. However, finding a threshold that optimizes either combined sensitivity and specificity or combined PPV and NPV is challenging.

If one has access to a partial gold-standard, it may be feasible to optimize the threshold for the combined sensitivity and specificity because this metric often has a single (global) minimum (see Fig. S1 in the Online Supplement). In general, this result follows from the fact that sensitivity and specificity are monotonically decreasing and increasing functions of the threshold, respectively. Thus, one can apply a greedy search procedure to find a threshold value corresponding to the optimal combined sensitivity and specificity.

However, the combined PPV and NPV and in general all metrics that incorporate PPV and NPV do not increase or decrease monotonically with the threshold (see Online Supplement for an explanation). Fig. S2 in the Online Supplement depicts the highly oscillatory nature of the combined PPV and NPV metric as a function of the threshold. In this case, a greedy search procedure that has access to a partial gold-standard would only find a local minimum.

On the other hand, if one does not have access to a partial gold-standard, finding an optimal threshold is infeasible for both combined sensitivity and specificity, and combined PPV and NPV metrics. These nuances in the interpretation of metric-specific performance are critical for practical applications of the methods.

3. Discussion

This benchmarking study shows the absolute and comparative performance of 32 network reverse-engineering methods/variants in 15 genome-scale real and simulated datasets/gold-standards using several metrics for assessing the accuracy of reverse-engineering. The methods used in this study include a broad array of state-of-the-art algorithms from bioinformatics and systems biology. In addition, algorithms from quantitative disciplines such as statistics and computer science were used. The results of this study show that some methods need to be substantially improved upon, while others should be used routinely. Those that should be improved are Aracne, Relevance Networks 1, Hierarchical Clustering, Graphical Lasso, GeneNet, and MI 1. The following methods should be routinely used: CLR, SA-CLR, *qp*-graphs, LGL-Bach, Relevance Networks 2, Fisher, and MI 2. Among the latter group of methods are LGL-Bach and *qp*-graphs, both of which are state-of-the-art techniques from computer science that deserve routine use in network inference tasks in bioinformatics and systems biology.

Our results also show that several univariate methods provide a "gatekeeper" performance threshold that should be used when method developers assess the performance of their novel algorithms. Furthermore, our analysis highlights the difficulty in determining optimal thresholds for algorithms that output scores for network edges rather than a network graph. The thresholds reported in primary publications of the score-based methods may be overfitted to the specific datasets used and therefore may not be universally applicable. Moreover, our results show that there is often no systematic way of searching for the best threshold over various performance metrics. Finally, our findings articulate the need for comprehensive benchmarking studies of future network reverse-engineering algorithms.

3.1. Comparison to prior research in evaluation of network reverse-engineering algorithms

The need for a comprehensive evaluation of reverse-engineering algorithms is well understood by the scientific community. This led to the formation of the DREAM project—a series of four competitions designed to assess the accuracy of network reverse-engineering [40,45,46]. With only one exception, none of the DREAM challenges addressed the specific problem of de-novo reverse-engineering of genome-scale regulatory networks from real non-longitudinal observational microarray data. Instead, the challenges used data that was in-silico, non-genome-scale, and/or from longitudinal or controlled experiments. Moreover, the data often incorporated partial biological knowledge. Thus, the findings of the DREAM challenges are outside the scope of this work and of many practical applications of reverse-engineering methods in real datasets. An exception is the DREAM2 challenge that included a task to reverse-engineer a network from a single *E. coli* microarray dataset.[3] Six algorithms were submitted, and the best performing method SA-CLR was considered to be a winner. However, as was mentioned previously, a winning performance in a single dataset may be a result of overfitting. Thus, one really has to assess algorithms over several datasets to reach reproducible conclusions. In addition to using 15 gold-standards/datasets in our study, we improve on the DREAM2 genome-scale challenge by using more methods for reverse-engineering, including newer methods that either were not available at the time of the DREAM2 challenge or did not participate in that competition.

To investigate the possibility of overfitting of SA-CLR to DREAM2 results, we included this method and the original DREAM2 *E. coli* dataset (labeled as "ECOLI(D)") in our evaluation and obtained the same AUPR and AUROC scores as in the DREAM2 challenge (see results for the ECOLI(D) dataset in Tables 3 and 4). However, SA-CLR was not a top-5 method across all 15 gold-standards/datasets in our study according to the AUROC metric (Table 5). This suggests possible overfitting of this method to the DREAM2 dataset and highlights the need for multiple datasets in the evaluation of methods.

It is also worthwhile mentioning the study of Bansal et al. who performed an evaluation of reverse-engineering methods on both real and simulated microarray datasets and ran algorithms de-novo using non-longitudinal observational data [6]. Our work significantly extends this prior work. First, the authors of that work assessed only 2 methods on real non-longitudinal genome-scale data, whereas our study compared 32 methods/variants. Second, the work of Bansal et al. [6] involved only 2 gold-standards of genome-scale sizes: one for the Yeast regulatory network [31] and the other for the 26-gene local pathway of *MYC* gene [8]. However, the latter gold-standard is incomplete (see Online Supplement), whereas the former one is

[3] To be precise, this DREAM2 challenge was not completely de-novo because a list of 152 transcription factors was given to each participant.

Table 5
Final ranking of methods according to each of the four performance metrics (benchmarks). The top-5 ranking methods for each benchmark are highlighted with red; other methods are highlighted with blue. Methods that are top-5 performers in at least one of the four benchmarks are considered to be "best of breed". Details about methods and their parameters can be found in Table 8 and Table S3 in the Online Supplement.

Method		*Final ranking according to*	
		combined sensitivity and specificity	**combined PPV and NPV**
Aracne	$\alpha = 10^{-7}$	12	8
	$\alpha = 0.05$	13	10
Relevance Networks 1	$\alpha = 10^{-7}$	7	20
	$\alpha = 0.05$	28	28
Relevance Networks 2		1	19
SA - CLR	$\alpha = 0.05$	4	16
	FDR = 0.05	11	11
CLR	*Normal MI estimator;* $\alpha = 0.05$	5	14
	Normal MI estimator; FDR = 0.05	10	9
	Stouffer MI estimator; $\alpha = 0.05$	2	15
	Stouffer MI estimator; FDR = 0.05	8	12
LGL-Bach	*max-k = 1, w/o symmetry*	17	5
	max-k = 2, w/o symmetry	20	4
	max-k = 3, w/o symmetry	19	3
	max-k = 1, with symmetry	18	6
	max-k = 2, with symmetry	21	2
	max-k = 3, with symmetry	22	1
Hierarchical Clustering		14	18
Graphical Lasso		15	7
GeneNet	$\alpha = 0.05$	9	17
	FDR = 0.05	16	13
qp - graphs	*q = 1*	24	23
	q = 2	25	26
	q = 3	27	27
	q = 20	26	25
	q = 200	23	23
Fisher	$\alpha = 0.05$	6	22
	FDR = 0.05	3	21
Full Graph		29	29
Empty Graph		29	29

Method		*Final ranking according to*	
		AUROC	**AUPR**
SA - CLR		7	4
CLR	*Normal MI estimator*	2	2
	Stouffer MI estimator	3	1
GeneNet		11	8
qp-graphs	*q = 1*	12	12
	q = 2	10	11
	q = 3	9	10
	q = 20	4	9
	q = 200	1	3
Fisher		8	6
MI 1		6	7
MI 2		5	5

outdated and not comprehensive in comparison with the most recent version of the Yeast regulatory map used in our evaluation [33]. Third, unlike [6], the synthetic gold-standards and data used in our study were generated to resemble real biological data (see Materials and methods section), and can therefore provide better estimates of anticipated performance of the methods in real data. Lastly, our study

Table 6
Description of the real gold-standards used in this study, along with the gene-expression data used for reverse-engineering the transcriptional network. See text for detailed description of gold-standards and datasets.

Dataset	Gold-standard				Gene expression data		
	Description	No. of TFs	No. of genes	No. of edges	Description	No. of arrays	No. of genes
ECOLI(A)	TF–gene interactions from RegulonDB 6.4 (strong evidence), [24]	140	1,053	1,982	*E. coli* gene expression dataset from Faith et al. [20]	907	4,297
ECOLI(B)	TF–gene interactions from RegulonDB 6.4 (strong and weak evidence), [24]	174	1,465	3,399			
ECOLI(C)	DREAM2 TF–gene network from RegulonDB 6.0, [46]	152	1,135	3,070			
ECOLI(D)	DREAM2 TF–gene network from RegulonDB 6.0, [46]	152	1,146	3,091	*E. coli* gene expression dataset from DREAM2 [46]	300	3,456
YEAST(A)	TF–gene interactions from [33], ($\alpha = 0.001$, C = 0)	116	2,779	6,455	Yeast gene expression dataset from Faith et al. [20]	530	5,520
YEAST(B)	TF–gene interactions from [33], ($\alpha = 0.001$, C = 1)	115	2,295	4,754			
YEAST(C)	TF–gene interactions from [33], ($\alpha = 0.001$, C = 2)	115	1,949	3,667			
YEAST(D)	TF–gene interactions from [33], ($\alpha = 0.005$, C = 0)	116	3,508	10,915			
YEAST(E)	TF–gene interactions from [33], ($\alpha = 0.005$, C = 1)	115	2,872	7,491			
YEAST(F)	TF–gene interactions from [33], ($\alpha = 0.005$, C = 2)	115	2,372	5,448			

utilized a suite of more sophisticated and informative performance metrics than sensitivity and PPV in order to evaluate the output of reverse-engineering algorithms from multiple perspectives.

Other recent efforts in comprehensive evaluation of reverse-engineering methods are typically limited to simulated data with a small number of genes, e.g. [26].

3.2. How accurately can the employed gold-standards be inferred from real gene expression microarray data?

Despite our rigor in using the correct implementation/application of each method and the most comprehensive gold-standards available to date, there are currently limits to the predictive power of reverse-engineering methods. Suppose there exists an optimal algorithm that could accurately discover all existing regulatory mechanisms from data using tests of statistical independence/association or functional equivalents. Unfortunately, the network produced by this algorithm would remain different from our gold-standards for a number of reasons. First, the Yeast and *E. coli* gold-standards are largely produced from experiments that show the physical binding of a transcription factor (TF) to DNA. However, such a binding event often does not lead to a functional change in gene expression, and hence one may not observe a corresponding statistical dependence in the microarray data [32]. Second, the regulatory network learned will likely be significantly dependent on the set of microarray experiments available. Often a transcription factor will affect the expression of different sets of genes in a condition-dependent manner. If certain TF–gene interactions only occur under conditions that are missing or underrepresented in the data, then a significant statistical dependence between the variables will be "drowned out" by the other samples. Third, some TFs or genes may have inherently low expression values that cannot be measured accurately by microarrays. They might be normalized out, as small changes in expression could be masked by noise in the data. Fourth, cellular aggregation and sampling from mixtures of distributions that are abundant in microarray data can also hide some statistical independence/association relations [15]. The above limitations are not specific to reverse-engineering algorithms, but are specific to assays and experimental design. Therefore, the performance results obtained in our study can be considered lower bounds on performance achievable by these algorithms.

3.3. On performance metrics for assessing accuracy of network reverse-engineering algorithms

Since there is no single context-independent metric to assess the accuracy of reverse-engineering methods, we used four different metrics to evaluate the results from different "angles." The combined PPV and NPV represents a measure of how precisely positive and negative interactions are predicted. The combined sensitivity and specificity favors methods that find an equally balanced trade-off between false-positive and false-negative interactions. AUROC represents the probability that a method ranks a true edge higher than a false edge, and hence quantifies the classification power of an algorithm. Finally, AUPR provides threshold-independent assessment of both the completeness (recall) and precision of a method.

One of the goals of reverse-engineering methods is to present experimentalists with a manageable list of putative regulatory interactions associated with a biological context. Many experimentalists are only concerned with the pathway (local network) around a single transcription factor, while others may be interested in broader network motifs. As a result, certain statistical metrics are more

Table 7
Description of the simulated gold-standards and dataset used in this study. See text for detailed description of gold-standards and datasets.

Dataset	Gold-standard				Gene expression data		
	Description	No. of TFs	No. of genes	No. of edges	Description	No. of arrays	No. of genes
REGED	REGED network	–	1,000	1,148	First 500 instances from REGED dataset	500	1,000
GNW(A)	Yeast regulatory network from GNW 2.0	157	4,441	12,864	25 time series with 21 time points in each generated by GNW 2.0	525	4,441
GNW(B)	1000-gene subnetwork of Yeast regulatory network from GNW 2.0	68	1,000	3,221	25 time series with 21 time points in each generated by GNW 2.0	525	1,000
GNW(C)	*E. coli* network from GNW 2.0	166	1,502	3,476	25 time series with 21 time points in each generated by GNW 2.0	525	1,502
GNW(D)	1000-gene subnetwork of *E. coli* regulatory network from GNW 2.0	121	1,000	2,361	25 time series with 21 time points in each generated by GNW 2.0	525	1,000

applicable to a specific biological context than others. For example, a biologist with limited resources who is interested in learning a pathway should use a method that scores well with respect to the combined PPV and NPV metric, such as LGL-Bach. On the other hand, biologists more interested in general regulatory patterns in a network, or with the resources to perform large-scale silencing experiments or binding analysis, might be more interested in an algorithm that scores well with respect to the combined sensitivity and specificity or AUROC metric. Part of our ongoing work is to elucidate the biological context specificity of reverse-engineering algorithms and derive context-specific performance metrics.

4. Materials and methods

4.1. Real datasets and gold-standards

Real gold-standards and microarray datasets were obtained for both Yeast and *E. coli*. The Yeast gold-standard was built by identifying the promoter sequences that are both bound by TFs (according to ChIP-on-chip data) and conserved within the *Saccharomyces* genus [28,33]. Binding information is essential because TFs must first bind to a gene to induce or suppress expression, while conservation information is important because true-positive TF–DNA interactions are often conserved within a genus. This study used a set of Yeast gold-standard networks that ranged from conservative to liberal. To obtain this range, networks were chosen with different ChIP-on-chip binding significance levels $\alpha = 0.001$ or 0.005, and were required to have a binding sequence that is conserved in $C = 0$, 1 or 2 of the related *Saccharomyces* species (Table 6). Hence, the most conservative gold-standard, YEAST(C), was built from TF–DNA interactions with $\alpha = 0.001$, such that bound DNA sequence was conserved in at least 2 Yeast relatives. A compendium of 530 Yeast microarray samples was taken from the Many Microbe Microarray Database [20].

The *E. coli* gold-standard network was obtained from RegulonDB (version 6.4), a manually curated database of regulatory interactions obtained mainly through a literature search [24]. ChIP-qPCR data has

Table 8
The list of reverse-engineering methods along with a brief description, computational complexity, and references. Methods denoted with "†" can only output graphs, and were therefore assessed only with combined sensitivity and specificity and combined PPV and NPV metrics. Methods denoted with "*" were assessed with the above metrics by converting their output (scores for all graph edges) into a graph by thresholding edge scores at the significance levels stated in Table S3 in the Online Supplement; these methods were also assessed with AUROC and AUPR without thresholding their output. Methods denoted with "#" by design only score edges, and were therefore assessed with AUROC and AUPR metrics. While *qp*-graphs are listed in the multivariate causal graph-based category, they can also be considered a representative of the multivariate Gaussian-graphical models family. The "Complexity" column has the following notation: p = number of genes in the dataset; n = number of samples in the dataset (typically, $n >> p$); q = size of conditioning set in *qp*-graphs; r = number of conditional independence tests performed for each pair of genes in *qp*-graphs; $m = max\text{-}k$ parameter of LGL-Bach that denotes maximum size of a conditioning set; $|PC|$ = average number of genes in the local causal neighborhood (i.e., genes directly upstream and downstream of the target gene).

Algorithm	Brief description	Complexity	References
(I) Univariate			
Relevance Networks 1†	Genes with statistically significant pairwise mutual information (MI) are connected by edges. MI was estimated using the procedure from Aracne algorithm.	$O(n^2p^2)$	[8,10,36]
Relevance Networks 2†	Genes with statistically significant pairwise mutual information (MI) are connected by edges. MI was estimated using the procedure by Qiu et al. This method incorporated the procedure of Butte and Kohane to assess significance of gene pairwise MI.	$O(n^2p^2)$	[10,41]
CLR*	Genes with statistically significant mutual information (MI) relative to background MI are connected by edges.	$O(n^2p^2)$	[21]
Fisher*	Genes with statistically significant association according to Fisher's *Z*-test are connected by edges.	$O(np^2)$	[4]
MI 1#	All possible gene edges are scored according to the strength of pairwise mutual information (MI). MI was estimated using the procedure from Aracne algorithm.	$O(n^2p^2)$	[8,10,36]
MI 2#	All possible gene edges are scored according to the strength of pairwise mutual information (MI). MI was estimated using the procedure by Qiu et al.	$O(n^2p^2)$	[41]
(II) Multivariate mutual information-based			
Aracne†	First, genes with statistically significant pairwise mutual information are connected by edges. Next, data processing inequality (DPI) is applied to triplets of genes in order to eliminate indirect interactions.	$O(p^3 + n^2p^2)$	[8,10,36]
SA-CLR*	Uses three-way mutual information to score triplets of genes, and connect genes by edges assuming cooperative regulation.	$O(n^3p^3)$	[48]
(III) Multivariate correlation-based			
Hierarchical Clustering†	Clustering genes by pairwise Pearson correlation. Genes in each clique are connected by edges.	$O(p^3 + np^2)$	[6,19]
(IV) Multivariate causal graph-based			
LGL-Bach†	Causal graph-based method based on (i) learning unoriented graph (network skeleton) via application of HITON-PC to all genes and (ii) orientation/pruning using Bach's scoring metric.	$O(p^2\|PC\|^m(m^2n + m^3))$ (for stage (i))	[2,3,5]
qp-graphs*	This algorithms starts from a full graph and searches for a subset of genes that renders two genes conditionally independent of each other.	$O(p^2r(q^2n + q^3))$	[12,13]
(V) Multivariate Gaussian graphical models			
GeneNet*	Edges are added between genes with non-zero full-order partial correlation. Correlations are found by estimating a covariance matrix using a shrinkage method.	$O(p^3 + np^2)$	[38]
Graphical Lasso†	Edges are added between genes with non-zero full-order partial correlation. Correlations are found by estimating a covariance matrix using coupled lasso regressions.	$O(p^3 + np^2)$	[23,37]

shown RegulonDB to be approximately 85% complete [46]. Evidence for each regulatory interaction in RegulonDB is classified as "strong" or "weak", depending on the type of experiment used to predict the interaction. For example, binding of a TF to a promoter is considered strong evidence, whereas gene-expression based computational predictions are considered weak evidence. For the purposes of our study, we created a conservative gold-standard of only strong interactions, and a liberal gold-standard that includes both strong and weak interactions. To ensure that our results are directly comparable with the DREAM2 challenge, we also included an earlier version of the RegulonDB gold-standard (see Table 6). A compendium of 907 *E. coli* microarray samples was taken from the Many Microbe Microarray Database [20]. We also used gene expression data from the DREAM2 challenge that was a subset of the previous dataset.

4.2. Simulated datasets and gold-standards

In addition to using real gene expression data with approximate gold-standards, we also used simulated data where gold-standards are known exactly (Table 7). We focused here exclusively on cutting-edge simulation methods that produce artificial data that resembles real biological data.

The Resimulated Gene Expression Dataset (REGED) is based on a high-fidelity resimulation technique for generating synthetic data that is statistically indistinguishable from real expression data [25,44]. The REGED network was induced from 1,000 randomly selected genes in a lung cancer gene expression dataset [9]. This network displays a power–law connectivity [30] and generates data that is statistically indistinguishable from real data according to an SVM classifier [47]. Moreover, statistical dependencies and independencies are consistent between the real and synthetic data according to the Fisher's Z test. Note that the REGED dataset was used in the Causality and Prediction Challenge [25].

The GeneNetWeaver (GNW) simulation method attempts to mimic real biological data by using topology of known regulatory networks [34,35]. Stochastic dynamics that are meant to model transcriptional regulation were applied to the extracted networks to generate simulated data.

4.3. Network reverse-engineering methods

This study used both univariate and four classes of multivariate network reverse-engineering methods: correlation-based, mutual information-based, causal graph-based and Gaussian graphical models (Table 8). While Aracne, Relevance Networks, LGL-Bach, Graphical Lasso, and Hierarchical Clustering output a graph (adjacency matrix), other methods output a symmetric matrix of scores that represent the relative likelihood of a regulatory interaction between any two genes. To obtain a graph for the latter methods, a threshold was chosen and an edge was formed between genes that have a score larger than the threshold. Methods MI 1 and MI 2 were used without thresholding, because otherwise they become equivalent to Relevance Networks that were already included in the study.

Since the problem of regulatory network reverse-engineering is NP-hard, only an algorithm that is worst-case exponential in the number of genes in the dataset can be both sound and complete[4] [14]. With the exception of LGL-Bach that is sound and complete,[5] all other algorithms used in this study have by design low-order polynomial complexity (Table 8) and therefore cannot possibly be sound and complete. Notably, LGL-Bach is not always exponential, but rather adjusts its complexity to the network that produced the data, and in many distributions runs faster than other tested methods.

We used the original author implementations of all methods except for Relevance Networks 2 and Fisher (see Table S3 in the Online Supplement). The original implementations for the latter two methods were not available, and we programmed them in Matlab. We used default author-recommended parameters for all methods whenever they were programmed in the software, stated in the original manuscript, or provided by the authors. In addition, we used popular statistical thresholds, as described in Table S3 in the Online Supplement. This allowed us to explore different configurations of the algorithms and assess their performance characteristics. All methods except for SA-CLR were run on a high performance computing facility in the Center of Health Informatics and Bioinformatics (CHIBI) at New York University Langone Medical Center. SA-CLR was run by its creators on a Columbia University cluster.

4.4. Performance assessment metrics

For the methods that directly output a network graph, we first computed positive predictive value (PPV), negative predictive value (NPV), sensitivity, and specificity. PPV measures the probability that a regulatory interaction discovered by the algorithm exists in the gold-standard (i.e., the precision of the output graph), while NPV measures the probability that an interaction *not* predicted by the algorithm does *not* exist in the gold-standard. Sensitivity measures the proportion of interactions in the gold-standard that are discovered by the algorithm (i.e., the completeness of the output graph), whereas specificity measures the proportion of interactions absent in the gold-standard that are *not* predicted by the algorithm. Then, PPV and NPV were combined in a single metric by computing the Euclidean distance from the optimal algorithm with PPV = 1 and NPV = 1: $\sqrt{(1-\text{PPV})^2 + (1-\text{NPV})^2}$. Similarly, we combined sensitivity and specificity by computing the Euclidean distance to the optimal algorithm with sensitivity = 1 and specificity = 1: $\sqrt{(1-\text{sensitivity})^2 + (1-\text{specificity})^2}$ [22]. These metrics take values between 0 and $\sqrt{2}$, where 0 denotes performance of the optimal algorithm and $\sqrt{2}$ denotes performance of the worst possible algorithm. A *smaller* value for either of these two metrics implies a more accurate algorithm.

For the methods that do not directly output a network graph, but rather output scores for the edges, we computed the area under the ROC curve (AUROC) and the area under the precision–recall curve (AUPR) [16,27]. These metrics take values between 0 and 1, where 0 denotes performance of the worst possible algorithm and 1 denotes performance of the optimal algorithm. For AUROC, 0.5 denotes performance of an algorithm that randomly scores edges. A *larger* value for either of these two metrics implies a more accurate algorithm.

We note that all of the above metrics were used in this study to measure the performance based on the *undirected* graphs output by each algorithm. Inference of *directed* graphs from data remains a more challenging problem that is beyond the scope of the present study.

4.5. Statistical analysis

The performance ranks of all algorithms were computed taking into consideration 95% confidence intervals around all point estimates. For example, if some method is the best performing one with AUROC = 0.98 and 95% confidence interval = [0.95, 1], then a method with AUROC = 0.96 is assigned the same rank as the best performing method. The confidence intervals were obtained using the methods of DeLong et al. [17] for AUROC; Richardson and Domingos [43] for AUPR; and the hyper-geometric test for combined sensitivity and specificity and combined PPV and NPV.

[4] An algorithm is considered "sound" if it outputs only true-positive gene-interactions. An algorithm is "complete" if it produces all true-positive gene-interactions, i.e. the entire network.

[5] LGL-Bach is provably sound and complete for learning the graph skeleton (undirected graph) [2,3].

Because the maximum rank may differ from dataset to dataset (e.g., due to ties), the obtained "raw" ranks were normalized and averaged over all datasets.[6] These average ranks were then used to obtain the final ranking of all methods (Table 5) according to the given performance metric.

Acknowledgments

Alexander Statnikov and Constantin F. Aliferis are acknowledging support from grants R56 LM007948-04A1 from the National Library of Medicine, National Institute of Health and 1UL1RR029893 from the National Center for Research Resources, National Institutes of Health. Varun Narendra was supported by the New York University Medical Science Training Program. We would also like to acknowledge Dimitris Anastassiou and John Watkinson for modifying the SA-CLR algorithm for our experiments and for running it on the Columbia University high performance computing facility; Boris Hayete for providing us with details on reproducing results of [21]; Robert Castelo for providing us with details on reproducing results of [13]; Peng Qiu for providing us with codes for fast computation of pairwise mutual information as in [41]; and Thomas Schaffter and Daniel Marbach for assistance with GeneNetWeaver gene network simulator.

Appendix A. Supplementary data

Supplementary data to this article can be found online at doi:10.1016/j.ygeno.2010.10.003.

References

[1] M.E. Adriaens, M. Jaillard, A. Waagmeester, S.L. Coort, A.R. Pico, C.T. Evelo, The public road to high-quality curated biological pathways, Drug Discov. Today 13 (2008) 856–862.
[2] C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, X.D. Koutsoukos, Local causal and Markov blanket induction for causal discovery and feature selection for classification: Part I. Algorithms and empirical evaluation, J. Mach. Learn. Res. 11 (2010) 171–234.
[3] C.F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, X.D. Koutsoukos, Local causal and Markov blanket induction for causal discovery and feature selection for classification: Part II. Analysis and extensions, J. Mach. Learn. Res. 11 (2010) 235–284.
[4] T.W. Anderson, An Introduction to Multivariate Statistical Analysis, Wiley-Interscience, Hoboken, NJ, 2003.
[5] F.R. Bach, M.I. Jordan, Learning graphical models with Mercer kernels, Adv. Neural Inf. Process. Syst. (NIPS) 15 (2003) 1009–1016.
[6] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, B.D. di, How to infer gene networks from expression profiles, Mol. Syst. Biol. 3 (2007) 78.
[7] T. Barrett, D.B. Troup, S.E. Wilhite, P. Ledoux, D. Rudnev, C. Evangelista, I.F. Kim, A. Soboleva, M. Tomashevsky, K.A. Marshall, K.H. Phillippy, P.M. Sherman, R.N. Muertter, R. Edgar, NCBI GEO: archive for high-throughput functional genomic data, Nucleic Acids Res. 37 (2009) D885–D890.
[8] K. Basso, A.A. Margolin, G. Stolovitzky, U. Klein, R. la-Favera, A. Califano, Reverse engineering of regulatory networks in human B cells, Nat. Genet. 37 (2005) 382–390.
[9] A. Bhattacharjee, W.G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E.J. Mark, E.S. Lander, W. Wong, B.E. Johnson, T.R. Golub, D.J. Sugarbaker, M. Meyerson, Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses, Proc. Natl Acad. Sci. USA 98 (2001) 13790–13795.
[10] A.J. Butte, I.S. Kohane, Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements, Pac. Symp. Biocomput. (2000) 418–429.
[11] J. Carrera, G. Rodrigo, A. Jaramillo, Towards the automated engineering of a synthetic genome, Mol. Biosyst. 5 (2009) 733–743.
[12] R. Castelo, A. Roverato, A robust procedure for Gaussian graphical model search from microarray data with p larger than n, J. Mach. Learn. Res. 7 (2006) 2650.
[13] R. Castelo, A. Roverato, Reverse engineering molecular regulatory networks from microarray data with qp-graphs, J. Comput. Biol. 16 (2009) 213–227.
[14] D.M. Chickering, D. Heckerman, C. Meek, Large-sample learning of Bayesian networks is NP-hard, J. Mach. Learn. Res. 5 (2004) 1287–1330.

[6] The ranks were first averaged over multiple versions of datasets (e.g., GNW(A), GNW(B), GNW(C), GNW(D)), and then the grand average was obtained over 4 dataset types (REGED, GNW, ECOLI, and YEAST).

[15] T. Chu, C. Glymour, R. Scheines, P. Spirtes, A statistical problem for inference to regulatory structure from associations of gene expression measurements with microarrays, Bioinformatics 19 (2003) 1147–1152.
[16] J. Davis, M. Goadrich, The relationship between precision–recall and roc curves, Proceedings of the 23rd international conference on Machine learning, 2006, p. 240.
[17] E.R. DeLong, D.M. DeLong, D.L. Clarke-Pearson, Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach, Biometrics 44 (1988) 837–845.
[18] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley, New York, 2001.
[19] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proc. Natl Acad. Sci. USA 95 (1998) 14863–14868.
[20] J.J. Faith, M.E. Driscoll, V.A. Fusaro, E.J. Cosgrove, B. Hayete, F.S. Juhn, S.J. Schneider, T.S. Gardner, Many Microbe Microarrays Database: uniformly normalized Affymetrix compendia with structured experimental metadata, Nucleic Acids Res. 36 (2008) D866–D870.
[21] J.J. Faith, B. Hayete, J.T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J.J. Collins, T.S. Gardner, Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles, PLoS Biol. 5 (2007) e8.
[22] L. Frey, D. Fisher, I. Tsamardinos, C.F. Aliferis, A. Statnikov, Identifying Markov blankets with decision tree induction, Proceedings of the Third IEEE International Conference on Data Mining (ICDM), 2003.
[23] J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics 9 (2008) 432–441.
[24] S. Gama-Castro, V. Jimenez-Jacinto, M. Peralta-Gil, A. Santos-Zavaleta, M.I. Penaloza-Spinola, B. Contreras-Moreira, J. Segura-Salazar, L. Muniz-Rascado, I. Martinez-Flores, H. Salgado, C. Bonavides-Martinez, C. Abreu-Goodger, C. Rodriguez-Penagos, J. Miranda-Rios, E. Morett, E. Merino, A.M. Huerta, L. Trevino-Quintanilla, J. Collado-Vides, RegulonDB (version 6.0): gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation, Nucleic Acids Res. 36 (2008) D120–D124.
[25] I. Guyon, C. Aliferis, G. Cooper, A. Elisseeff, J.P. Pellet, P. Spirtes, A. Statnikov, Design and analysis of the causation and prediction challenge, J. Mach. Learn. Res. Workshop Conf. Proc. 3 (2008) 1–33.
[26] H. Hache, H. Lehrach, R. Herwig, Reverse engineering of gene regulatory networks: a comparative study, EURASIP J. Bioinform. Syst. Biol. (2009) 617281.
[27] D.J. Hand, R.J. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems, Mach. Learn. 45 (2001) 171–186.
[28] C.T. Harbison, D.B. Gordon, T.I. Lee, N.J. Rinaldi, K.D. MacIsaac, T.W. Danford, N.M. Hannett, J.B. Tagne, D.B. Reynolds, J. Yoo, E.G. Jennings, J. Zeitlinger, D.K. Pokholok, M. Kellis, P.A. Rolfe, K.T. Takusagawa, E.S. Lander, D.K. Gifford, E. Fraenkel, R.A. Young, Transcriptional regulatory code of a eukaryotic genome, Nature 431 (2004) 99–104.
[29] C. Huttenhower, M.A. Hibbs, C.L. Myers, A.A. Caudy, D.C. Hess, O.G. Troyanskaya, The impact of incomplete knowledge on evaluation: an experimental benchmark for protein function prediction, Bioinformatics 25 (2009) 2404–2410.
[30] H. Jeong, B. Tombor, R. Albert, Z.N. Oltvai, A.L. Barabasi, The large-scale organization of metabolic networks, Nature 407 (2000) 651–654.
[31] T.I. Lee, N.J. Rinaldi, F. Robert, D.T. Odom, Z. Bar-Joseph, G.K. Gerber, N.M. Hannett, C.T. Harbison, C.M. Thompson, I. Simon, J. Zeitlinger, E.G. Jennings, H.L. Murray, D.B. Gordon, B. Ren, J.J. Wyrick, J.B. Tagne, T.L. Volkert, E. Fraenkel, D.K. Gifford, R.A. Young, Transcriptional regulatory networks in Saccharomyces cerevisiae, Science 298 (2002) 799–804.
[32] X.Y. Li, S. MacArthur, R. Bourgon, D. Nix, D.A. Pollard, V.N. Iyer, A. Hechmer, L. Simirenko, M. Stapleton, C.L. Luengo Hendriks, H.C. Chu, N. Ogawa, W. Inwood, V. Sementchenko, A. Beaton, R. Weiszmann, S.E. Celniker, D.W. Knowles, T. Gingeras, T.P. Speed, M.B. Eisen, M.D. Biggin, Transcription factors bind thousands of active and inactive regions in the Drosophila blastoderm, PLoS Biol. 6 (2008) e27.
[33] K.D. MacIsaac, T. Wang, D.B. Gordon, D.K. Gifford, G.D. Stormo, E. Fraenkel, An improved map of conserved regulatory sites for Saccharomyces cerevisiae, BMC Bioinform. 7 (2006) 113.
[34] D. Marbach, R.J. Prill, T. Schaffter, C. Mattiussi, D. Floreano, G. Stolovitzky, Revealing strengths and weaknesses of methods for gene network inference, Proc. Natl Acad. Sci. USA 107 (2010) 6286–6291.
[35] D. Marbach, T. Schaffter, C. Mattiussi, D. Floreano, Generating realistic in silico gene networks for performance assessment of reverse engineering methods, J. Comput. Biol. 16 (2009) 229–239.
[36] A.A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, F.R. Dalla, A. Califano, ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, BMC Bioinform. 7 (Suppl 1) (2006) S7.
[37] N. Meinshausen, P. Buhlmann, High-dimensional graphs and variable selection with the lasso, Ann. Stat. 34 (2006) 1436–1462.
[38] R. Opgen-Rhein, K. Strimmer, From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data, BMC Syst. Biol. 1 (2007) 37.
[39] H. Parkinson, M. Kapushesky, N. Kolesnikov, G. Rustici, M. Shojatalab, N. Abeygunawardena, H. Berube, M. Dylag, I. Emam, A. Farne, E. Holloway, M. Lukk, J. Malone, R. Mani, E. Pilicheva, T.F. Rayner, F. Rezwan, A. Sharma, E. Williams, X.Z. Bradley, T. Adamusiak, M. Brandizi, T. Burdett, R. Coulson, M. Krestyaninova, P. Kurnosov, E. Maguire, S.G. Neogi, P. Rocca-Serra, S.A. Sansone, N. Sklyar, M. Zhao, U. Sarkans, A. Brazma, ArrayExpress update—from an archive of functional genomics experiments to the atlas of gene expression, Nucleic Acids Res. 37 (2009) D868–D872.

[40] R.J. Prill, D. Marbach, J. Saez-Rodriguez, P.K. Sorger, L.G. Alexopoulos, X. Xue, N.D. Clarke, G. Altan-Bonnet, G. Stolovitzky, Towards a rigorous assessment of systems biology models: the DREAM3 challenges, PLoS ONE 5 (2010) e9202.
[41] P. Qiu, A.J. Gentles, S.K. Plevritis, Fast calculation of pairwise mutual information for gene regulatory network reconstruction, Comput. Meth. Programs Biomed. 94 (2009) 177–180.
[42] D.R. Rhodes, A.M. Chinnaiyan, Integrative analysis of the cancer transcriptome, Nat. Genet. 37 (Suppl) (2005) S31–S37.
[43] M. Richardson, P. Domingos, Markov logic networks, Mach. Learn. 62 (2006) 107–136.
[44] A. Statnikov, C.F. Aliferis, Analysis and computational dissection of molecular signature multiplicity, PLoS Computational Biol. 6 (2010) e1000790.
[45] G. Stolovitzky, D. Monroe, A. Califano, Dialogue on reverse-engineering assessment and methods: the DREAM of high-throughput pathway inference, Ann. N.Y. Acad. Sci. 1115 (2007) 1–22.
[46] G. Stolovitzky, R.J. Prill, A. Califano, Lessons from the DREAM2 challenges, Ann. N.Y. Acad. Sci. 1158 (2009) 159–195.
[47] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[48] J. Watkinson, K.C. Liang, X. Wang, T. Zheng, D. Anastassiou, Inference of regulatory gene interactions from expression data using three-way mutual information, Ann. N.Y. Acad. Sci. 1158 (2009) 302–313.