BIOINFORMATICS

Vol. 22 no. 14 2006, pages e539–e546
doi:10.1093/bioinformatics/btl204

Inferring Functional Pathways from Multi-Perturbation Data

Nir Yosef[1,*,†], Alon Kaufman[2,†] and Eytan Ruppin[1,3]

[1]School of Computer Science, Tel-Aviv University, Tel-Aviv, Israel, [2]Center of Neural Computation, Hebrew University, Jerusalem, Israel and [3]School of Medicine, Tel-Aviv University, Tel-Aviv, Israel

ABSTRACT

Background: Recently, a conceptually new approach for analyzing gene networks, the Functional Influence Network (FIN) was presented. The FIN approach uses the measured performance of a given cellular function under different multi-perturbations, to identify the main functional pathways and interactions underlying its processing. Here we present and study an iterative, extended version of FIN, the Functional Influence Network Extractor (FINE), which is specifically geared towards the accurate analysis of sparse cellular systems. We employ it to study a conceptually fundamental question of practical importance—how well should we know the system studied (such that we can predict its performance) so that we can understand its workings (*i.e.,* chart its underlying functional network)?

Results and Conclusions: The performance of FINE is studied in both simulated and biological sparse systems. It successfully obtains an accurate and compact description of the underlying functional network even with limited data, and outperforms FIN. We show that prior estimates of a system's functional complexity are instrumental in determining how much predictive knowledge is required to accurately chart its underlying functional network.

Availability: The FINE software is available for download at http://www.cns.tau.ac.il/resc.html

Contact: niryosef@post.tau.ac.il

INTRODUCTION

Which elements within a system are important for its performance? How do these elements influence the system's performance, and to what extent? Are there inter-element interactions which significantly affect the system's performance? These fundamental questions typically arise when attempting to analyze a system in order to understand its workings. Specifically, within the context of genetic networks, the success of genome sequencing projects and high throughput gene expression studies has allowed biologists to identify almost all genes responsible for producing the biological complexity of several organisms. The next important task is to quantify their importance to various cellular functions (Carpenter *et al*., 2004) and understand their functional regulatory interactions (Barabasi *et al*., 2004) and the 'logic circuitry' (Davidson *et al*., 2002).

To causally deduce the roles played by genes in determining a cellular function, or more generally the role of elements in any system, perturbation studies are necessary and have been traditionally employed. In perturbation studies, phenotypic variation is traced after deletion or mutation of different genes. Nevertheless, the vast majority of these studies have employed single perturbations, which often result in little phenotypic effect, due to the existence of duplicates, alternative pathways and functional overlap (Gu *et al*., 2003). Hence, multiple concomitant perturbations should be employed in order to identify the causal contributions of the different genes to the system's functioning (Kaufman *et al*., 2005). Such studies have been quite scarce up until now, comprised of either large-scale studies of double knockouts (Tong *et al*., 2004), or small-scale studies spanning a broader span of multiple-knockouts (Yuh *et al*., 2001; Kaufman *et al*., 2004). However, the growing awareness that multi-perturbation studies are essential for deciphering the workings of complex genetic networks, along with the recent development of new experimental methods such as RNA interference (RNAi) (Hammond *et al*., 2001) and transposon mutagenesis, will soon lead to the accumulation of large amounts of multi-perturbation genetic data.

The goal of the algorithm at the basis of this paper is to reveal the main functional pathways and functional interactions uncovered by multiple knockout experiments in a genetic network. Obviously, there have been many studies which have developed methods to uncover the network of interactions between genes, mostly based on microarray data. Such studies typically address microarray data analysis by inferring regulatory networks using Boolean networks (Ideker *et al*., 2000), Bayesian networks (Pe'er *et al*., 2001) and other approaches (Ideker *et al*., 2001; Tegner *et al*., 2003). Unlike these approaches, our goal is to obtain causal functional descriptions, by analyzing data gathered in studies where a specific cellular function is probed using a variety of multiple knockout experiments. The functional interactions and pathways we aim to reveal do not necessarily imply any physical or direct biochemical interactions, and rather represent functional modules. Keinan *et al*. (2004) and Kaufman *et al*. (2005) have previously presented two complementary methods to address this challenge, and applied them to the analysis of genetic and neuronal multi-perturbation data. The latter presented the Functional Influence Network (FIN) algorithm, aiming to produce a *Compact Functional Network (CFN)* which describes in a compact and accurate manner how the genes, acting together in functional pathways, determine a certain cellular function or phenotypic behavior. In this study we expand the basic FIN approach in three fundamental ways:

(1) First, we develop a new algorithm—the Functional Influence Network Extractor (FINE), motivated by the empirical observation that many biological networks are functionally sparse

*To whom correspondence should be addressed.
†These authors made an equal contribution to this work.

(*e.g*. Thieffry *et al*., (1998); Jeong *et al*., (2000)), *i.e*. have a compact *functional backbone*.

(2) Second, we perform an extensive study of the workings of FINE. To this end, a comprehensive set of measures was developed to evaluate the performance of the algorithm, on a large number of simulated networks. We then applied FINE to the analysis of the *cis*-regulatory system of the sea urchin *endo16* gene (Yuh *et al*., 2001).

(3) The third contribution that this paper makes is conceptual: since obviously one cannot expect to obtain all possible multi-knockout experiments, a question arises: How many experiments will be needed to successfully identify the CFN which accurately describes the functioning of the system? Utilizing FINE, we address this question and study how its accuracy depends on the complexity of the system in hand.

METHODS

Algorithm Background: the FIN Approach

Experimental data obtained in multi-perturbation studies can give rise to two different kinds of knowledge: (i) *Predictive knowledge*—where given a new, unseen state of the system in hand (a new multi-knockout configuration) one can predict its functioning level, and (ii) *Descriptive knowledge*—where one attempts to reconstruct the functional backbone of the system, *i.e*. describe how the system's components actually interact to perform the function in question. It is the latter kind of knowledge which is the goal of FIN. To this end, it is composed of two parts: (i) Constructing a functional model, which describes how the elements in the system (genes) interact to determine the studied phenotypic behavior, and (ii) Simplifying the resulting functional model (which tends to be very large and unintelligible) and producing a compact, yet accurate, functional description of the system in hand, the CFN.

Constructing a Functional Model from Multi-Perturbation Data Let the investigated system be defined by a pair (N, F). $N = \{1, \ldots, n\}$ is the set of all elements in the system, where each element can be in one of two states, either intact(1) or perturbed(0). $F : \{0, 1\}^n \rightarrow R$, the performance function, associates to every set $S \subseteq N$ a number describing the performance level of the system when the set of elements S is intact, $S = \{x \in N \mid \text{state}(x) = 1\}$. For example, in genetic multi-knockout experiments, N denotes the set of all genes, and for each $S \subseteq N$, $F(S)$ denotes the quantitative phenotype measured in the knockout experiment in which all the genes in S are intact and the rest are knocked-out. A fundamental result from Game Theory shows that $F(S)$ can be uniquely decomposed into the sum $\sum_{T \subseteq S} a(T)$ (Grabisch *et al*., 2000), where the coefficients $a(T)$, denoted *dividends*, describe the marginal contribution of each subset T of the set of intact elements S to the studied performance function F. The dividends are calculated based on the performance levels measured in the different multi-knockout experiments, according to

$$a(S) = \sum_{T \subseteq S} (-1)^{|T|-|S|} F(T), \quad \forall S \subseteq N, \tag{1}$$

(where $|S|$ and $|T|$ denote the cardinality of the sets S and T respectively).

In the context of a data set of multi-knockout experiments with their associated measures performance levels, the dividend computation begins from the dividend of the null group, $a(\emptyset) = F(\emptyset)$ (the performance measured when all the elements are knocked out), and each iteration of Eq.(1) computes the dividend (marginal contribution) of the subsequent supersets. That is, in the second iteration the performance of the single elements minus the performance of the null group is computed, resulting in the marginal contribution of each single element. The third iteration computes the performance of the elements-pairs minus the performance of single elements plus the null group performance, resulting in the marginal contribution of each of the elements-pair, and so forth. Based on the dividends, the performance function F can be represented as a multi-linear polynomial:

$$F(\vec{x}) = \sum_{S \subseteq N} a(S) \cdot \prod_{i \in S} x_i \tag{2}$$

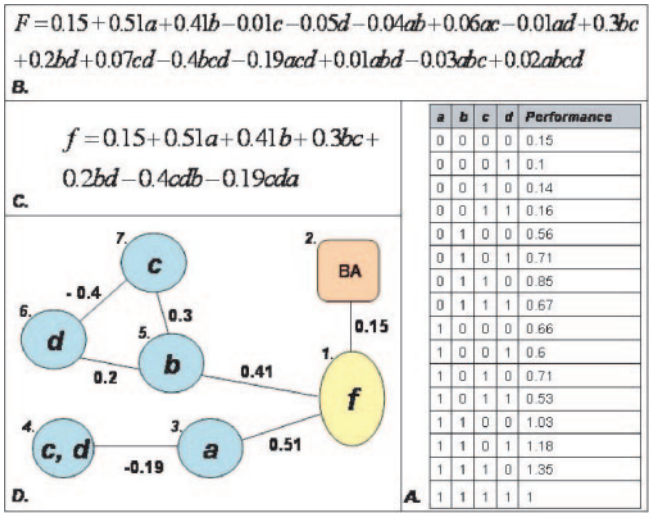
where the vector $\vec{x} \in \{0, 1\}^n$ describes the (intact/knocked-out) states of the elements in the system. Each term in the polynomial, denoted as *summand*, describes a distinct functional pathway since its elements must all be intact to influence the value of F. Obviously if the function is elementary, that is, if there are no dependencies between the elements, it could be fully approximated by a summation over the individual contributing elements (based on n single-knockout experiments). However, in the context of biological systems, such a description is likely to be insufficient and even misleading since such systems are usually complex and involve higher-order interactions.

Constructing the Compact Functional Network (CFN) In the practical analysis of genetic biological data, the full functional description of Eq.(2), is typically very large and unintelligible, containing many 'uninteresting' pathways with very small (but non-zero) influence (dividend). To address this problem, Kaufman *et al*. (2005) introduced the concept of the CFN, a compact representation which approximates the full functional description. The CFN is in itself a multi-linear polynomial which preserves only the most important summands of the full representation.

Figure 1 shows a schematic example of a CFN construction; the full set of all 2^n ($n = 4$) possible multi-knockout experiments is given in box *A*. This set yields a unique performance function F, describing the phenotypic behavior of the system (box *B*). The resulting CFN approximating the full functional description is shown in box *C*. With this compact CFN representation, the approximated performance function f can be visualized in a relatively simple graph (box *D*). This graph, referred to as the *functional diagram*, provides both predictive knowledge, acting as an oracle for the system's behavior at any given state, and descriptive knowledge—explicitly describing the functional structure of the system. Each node in the graph corresponds to a set of (possibly only one) elements and is said to be intact if, and only if, all of its corresponding elements are intact. Additional nodes are the basal activity node *BA* which corresponds to the empty set and the *output node f*. Each simple path which ends at the output node defines a functional pathway (a summand in the CFN) whose elements are those listed on the nodes along the path. The dividend of each such functional pathway is the weight on its first edge. For example, the dividend of the functional pathway (*c-d-b-f*) is the weight on the edge between nodes 7 and 6. Given a knockout experiment, the expected performance level of the CFN can be calculated by summing up the dividends of all the intact functional pathways, that is, sum up the weights on the edges between intact nodes which form a connected component with the output node (for an illustrative example, see legend of figure 1). Note that the existence of an edge between two nodes in the functional diagram does not necessarily imply that they are connected by any physical interaction. Instead, it denotes that there exists a summand in the CFN which contains both these elements, that is, they both participate in a joint functional pathway.

Evidently, the construction of the CFN requires the performance values over all possible multi-knockout experiments; producing such data is an unrealistic demand in most cases. In order to construct a CFN given partial multi-knockout data, the FIN algorithm (Kaufman *et al*., 2005) predicts the performance levels of the missing knockout experiments (using any desired prediction method) and computes the functional model (Eq.(2)) based on these predicted values. It then applies a pruning procedure to remove summands from the functional model, aiming to sustain only the most important ones, while maintaining a pre-defined level of accuracy (comparing the pruned model to the original functional model)[1]. The pruning process is

[1]Throughout the paper, when measuring the accuracy between two continuous vectors p and q, we report the percentage of the variance of p explained by q, this is, $100(1 - (\|p - q\|^2)/(\|p - \bar{p}\|^2))$ where $\bar{p}$ is the mean of p.



a	b	c	d	Performance
0	0	0	0	0.15
0	0	0	1	0.1
0	0	1	0	0.14
0	0	1	1	0.16
0	1	0	0	0.56
0	1	0	1	0.71
0	1	1	0	0.85
0	1	1	1	0.67
1	0	0	0	0.66
1	0	0	1	0.6
1	0	1	0	0.71
1	0	1	1	0.53
1	1	0	0	1.03
1	1	0	1	1.18
1	1	1	0	1.35
1	1	1	1	1

Fig. 1. *A simple schematic CFN construction of a* 4 *element system.* Box *A* provides the analyzed data set of the 16 (2^4) possible combinations of multi-knockout experiments and their corresponding performance measures. Box *B* shows the performance function *F* derived by a dividend analysis of the multi-knockout data set. For example, the dividend value of the subset $\{a, b\}$ is calculated as: Performance($\{a,b\}$) − Performance($\{a\}$) − Performance($\{b\}$) + Performance($\emptyset$) = 1.03 − 0.66 − 0.56 + 0.15 = − 0.04. Box *C* presents the resulting CFN. Finally, box *D* depicts its network visualization. Each of the round-shaped nodes (numbered 3 − 7) corresponds to a set of elements. Node 2 corresponds to the empty set, describing the system's basal activity. Node 1 is the output node. Given a knockout experiment where (for example) only *a*, *c* and *d* are intact then the intact functional pathways are (*a*-*f*) and ([*c*, *d*]-*a*-*f*), and the value of *f* is 0.51 − 0.19 + 0.15 = 0.47, the sum of dividends of intact pathways and the basal activity.

composed of selecting statistically significant summands, and then eliminating those remaining summands which have a low dividend magnitude. The output is the pruned polynomial, composed of the remaining summands.

The FINE Algorithm

Motivation While designed to construct a CFN as accurately as possible, the FIN algorithm still quite frequently produced lengthy and cumbersome descriptions when supplied with biological experimental data. The FINE algorithm was developed to overcome these pitfalls of the FIN for sparse biological systems. Such systems are characterized by an actual small number of important functional pathways in relation to the set of all pathways made possible by different groupings of their elements. Hence, the end goal CFN describing the working of these systems should be small as well. Based on the FIN as a building block, and taking the assumption that the system in hand is sparse, FINE performs an iterative process of pruning and re-approximation of the functional model and produces increasingly more compact and accurate CFN models.

Formal Description of the FINE Algorithm Given a system of n elements, the input of FINE is a set of q multi-knockout experiments with their corresponding experimental performance values (usually, $q \ll 2^n$). FINE is composed of a preprocessing phase of prediction, followed by an iterative process of CFN construction. It identifies the set of important summands and outputs an accurate CFN.

Preprocessing—Constructing the Full Data Predictor. We train a predictor on the accessible, incomplete data set to predict the performance levels of the missing experiments and compute the ensuing dividends of all the 2^n summands (Eq.(2)). This process is done using bootstrapping by randomly resampling with replacement from the available data. Any desired predictor can be used in this process, producing a pair (*D*, *PER*) as output. *D* is a $2^n \times B$ matrix, where $D_{i,j}$ is the estimated dividend of the i^{th} summand according to the j^{th} bootstrap repetition (*B* is the number of bootstrap repetitions), and $PER \in R^{2^n}$, is the predicted performance levels in all possible knockout experiments (taken as the mean prediction over all *B* bootstrap repetitions). We refer to the accuracy of this preprocessing prediction as the *prediction accuracy*.

Iterative Construction of the CFN. This is an iterative process in which the set of summands included in the CFN (i.e. with a non zero coefficient\dividend) is gradually pruned and narrowed down. The input and output of each iteration are pairs of the form (*D*, *PER*), as defined above. On each iteration, the number of non-zero rows in *D* is monotonically reduced.
Each iteration step is comprised of two phases:

(1) *Summands selection phase*—For each summand, we use the corresponding row in *D* to calculate two indices: (i) The significance level of its dividend (based on a *t*-test where the null hypothesis is that the dividend magnitude is zero), and (ii) The expected magnitude of its dividend (taking the mean value). The most important summands are then chosen based on these indices, using forward selection and backward elimination procedures. These procedures are controlled by two pre-determined target levels of accuracy, $level_1 > level_2$[2]. Starting from an empty summand set (an empty CFN), we gradually add summands to the CFN, doing so by the order of their dividends' significance, until we reach a desired accuracy of $level_1$. Next, we apply backwards elimination on the resulting set of significant CFN summands, now eliminating summands by the order of their dividends' magnitude (starting from the small ones) until the lower limit of accuracy, $level_2$, is reached.

(2) *Dividends recomputation phase*—Let *m* denote the cardinality of the set of important summands as of the preceding summands selection phase. We fit each of the *m* chosen summands a new dividend coefficient according to the following model: $Q \cdot \vec{\delta} = \vec{y}$, where *Q* is a binary $q \times m$ matrix describing the partial set of biological knockout experiments in hand, defined as: $Q_{j,i} = 1$ iff $T_i \subseteq S_j$, where T_i is the set of elements included in the i^{th} important summand, and S_j is the subset of genes intact in the j^{th} experiment. $\vec{y}$ is the given $q \times 1$ vector of observed performance levels. The coefficients vector $\vec{\delta}$ is therefore the new estimated dividends vector. Clearly, the number of free variables in this model decreases in each iteration since the set of important summands is monotonically reduced. When the matrix *Q* does not have a full rank, there is obviously no unique solution. The particular basic solution chosen is determined using the QR factorization with column pivoting (Businger *et al*., 1965). An over determined equation set is typically reached after a small number of iteration steps (on our simulations, the majority of cases did not require more than 5 iteration steps to reach an over determined equation set). Repeating the calculation of $\vec{\delta}$ using bootstrapping results in a new set of dividends, *D*, from which *PER* is calculated and both serve as the input to the next iteration.

The iterative process continues until the following stopping criteria is satisfied: either the given model cannot be pruned (*i.e.,* the output of an iteration is equal to its input) or that a user defined upper bound on the number of iterations is reached. (the upper bound of 10 iterations, used in our simulations, was reached in approximately 1% of the experiments).

The algorithm returns the output of the last prediction phase: a predicted set of all 2^n knockout experiments (*PER*) and a multi-linear polynomial whose coefficients (most of them zero) are taken as the mean values over the rows of *D*. This is a CFN representation of the given performance

[2]The accuracy level is computed between the prediction based on the chosen dividends and the prediction given by the previous iteration.

function in which all remaining summands (those with non zero dividend coefficients) are important per our definition and whom cannot be further eliminated.

To visualize the output of FINE, we have developed an automated module for the construction of functional diagrams, based on a series of factorization steps applied on the CFN polynomial. Due to space limitations details are not provided, however, this module is available as a part of the FINE software package.

RESULTS

Measures for Evaluating FINE

We present a set of measures, testing to what extent does FINE achieve its objectives. These measures evaluate the accuracy of the CFN obtained using partial knockout data to that obtained with full data. We define: *Ground truth performance*—the vector of all 2^n knockout experiments' performance levels, as given by the predicted performance function. *Ground truth CFN (GTCFN)*—the CFN obtained by applying FINE to the full data set of *ground truth performance* values. The *CFN performance* is a vector of all 2^n performance levels computed from a given CFN over all possible knockout experiments. Based on these definitions we present the following measures to quantify CFN accuracy:

- **Operational accuracy**—the ability to produce accurate predictions of the system's behavior at any given state, measured by the match between the ground truth performance and the performance values predicted by the CFN.
- **Dividend accuracy**—the accuracy of the weights assigned to each functional pathway, measured by the match between all 2^n dividends (some are zero) of the CFN and the GTCFN.
- **Descriptive accuracy**—the ability to detect the most important pathways. Since the GTCFN, by construction, contains only the most important summands of the original target function (such that the pre-defined level of accuracy is satisfied) it can be used as a ''gold standard'' for measuring the descriptive accuracy. We therefore compare the CFN summands to the GTCFN summands through the following measures:
- *Specificity*—the total magnitude of CFN dividends whose corresponding summands appear in the GTCFN, divided by the total magnitude of all CFN dividends.
- *Sensitivity*—the total magnitude of GTCFN dividends whose corresponding summands are included in the CFN, divided by the total magnitude of all the GTCFN dividends.
- *Jaccard coefficient*—the number of CFN summands which appear in the GTCFN (tp), divided by the combined number of GTCFN summands (t) and the CFN summands which do not appear in the GTCFN (fp). This score reflects the 'conjunction over union' between the CFN summands and GTCFN summands, $(tp/(t+fp))$.
- *Top summands detection rate*—the success rate in identifying the three most important summands in a given performance function, where each summand is ranked proportionally to the number of multi-knockout experiments on which it affects. In a system of n elements, the rank of a summand with s elements and a dividend value of d is set to $|d| \cdot 2^{n-s}$.

Combined, these measures provide a comprehensive evaluation of the performance of FINE. Overall, FINE should give an accurate approximation of the actual function investigated, (*operational accuracy*), in which the important subsets of elements are expressed (*descriptive accuracy*) with the accurate weights assigned to them (*dividend accuracy*). Note that the *operational accuracy* is different from the *prediction accuracy* as the former relates to the preliminary prediction and the latter, to the output of FINE.

FINE Analysis of Simulated Data: Descriptive Vs. Predictive Accuracy

First, a comparison of FINE with FIN in the analysis of sparse systems is in hand. Figure 2 illustrates the continuous improvement in both descriptive accuracy and dividends accuracy throughout the iterative process of FINE, measured in our simulation experiments. Evidently, the more sparse the system is, the more significant is the improvement along the iterative process. These results clearly demonstrate the superiority of FINE over the FIN algorithm in sparse systems (as the FIN is equivalent to FINE with a single iteration).

Our main focus is to utilize FINE to attend the following fundamental questions: having obtained multi-knockout performance data of some cellular function, how well can we expect to understand and describe it's processing? In terms of this paper, how is the descriptive accuracy of a CFN produced by FINE dependent on the prediction accuracy of the data that has been collected? Furthermore, how and to what extent is this relation dependent on the architecture of the underlying network, *i.e.* the studied performance function? In biological systems, these underlying networks are currently mostly unknown. Therefore, the dependence of the operational and descriptive accuracy on the prediction accuracy is important, since prediction accuracy is the only measure one may have in hand. To study these questions in depth, we perform a comprehensive set of experiments using simulated multi-knockout performance data.

The Simulation Experiments We generate a set of random performance functions, each inducing a different functional backbone network architecture. These functions are multi-linear polynomials[3], parameterized by (n, m, c): having n elements, with m summands (functional pathways), each summand containing no more than c elements (the length of the pathways is bounded by c). We study a wide range of functions, varying form simple ($n = 8$, $m = 2$, $c = 2$) to more complex ($n = 8$, $m = 16$, $c = 8$). For each parameter set (n, m, c) we consider 10 random performance functions, each inducing a different network architecture. The coefficients of each polynomial are selected randomly from a uniform distribution (on the interval [6, 10]) and arbitrarily assigned with a ± sign. The input data to FINE is a set of 'knockout experiments' obtained by considering different intact subsets out of the n elements and calculating their corresponding performance levels. For each of the random performance functions, we performed a set of FINE analyses using a span of partial input data sets, ranging from 10 to 256 samples (out of $2^8 = 256$) yielding a wide range of prediction accuracies. In the current implementation we used k-nearest neighbors (KNN) as the 'default' prediction method (used in the preprocessing stage) with the parameter k set to 3. The target levels of accuracy, $level_1$ and $level_2$, were set to 98% and 95% respectively.

[3]Our choice of multi-linear polynomials as target models stemmed from the fact that performance levels of any multi-knockout data set can be uniquely described in such canonical form.

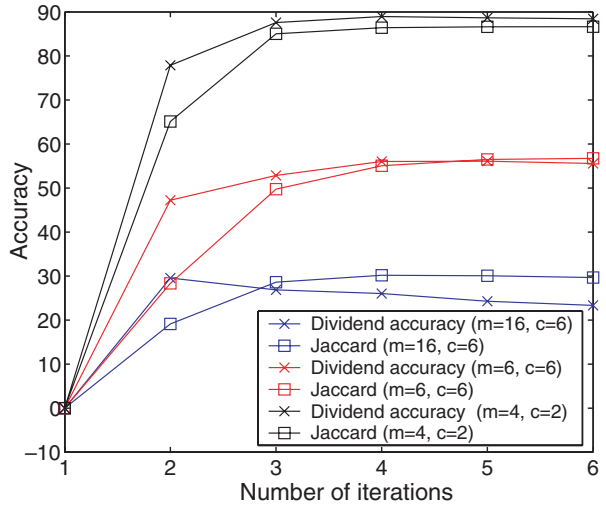


Fig. 2. *Convergence and improvement in accuracy across FINE selection and recomputation iterations.* The figure depicts the improvement of the Jaccard coefficient and the dividends accuracy (y-axis) across the algorithm's iterations (x-axis) until convergence. Data presented for a set of performance functions with three different parameter sets (detailed on the inset of the figure), where each parameter set defines a different level of sparseness (see next section). The continuous improvement in the Jaccard coefficient and dividend accuracy implies that we continuously eliminate more false positive (*fp*) summands than true positive (*tp*) ones (Jaccard coefficient = $tp/(t+fp)$) and that the dividend coefficients assigned to these summands are increasingly more accurate. The point of reference (on x = 1) is the result of the first iteration, or equivalently, the result of the FIN algorithm.

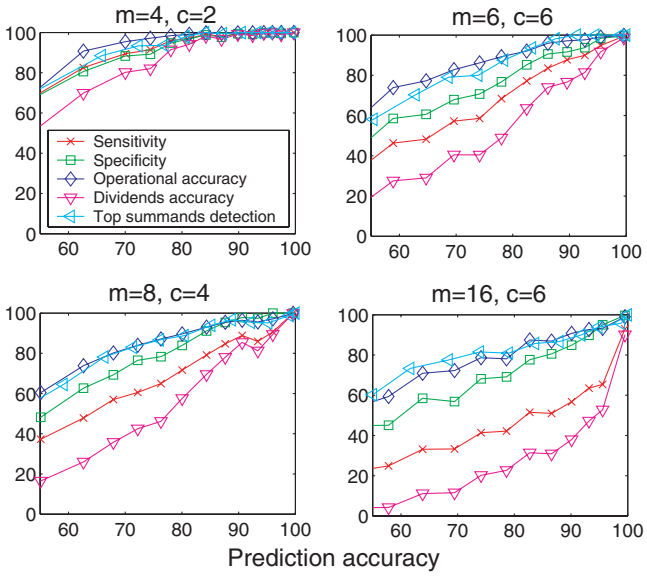


Fig. 3. *Performance of FINE on simulated data.* The different performance measures (y-axis) are plotted against the prediction accuracy of the input given to the algorithm (x-axis). Four different performance functions with different parameter sets are displayed.

The Simulation Results Figure 3 demonstrates the simulation results. The different performance measures are plotted against the prediction accuracy of the sample sets. Evidently, all the performance measures increase with the prediction accuracy. Notably, in the more complex functions, the descriptive accuracy measures rise to high levels only at fairly high prediction accuracy levels. This implies that when the assumed size and complexity of the biological system studied is considerable, one must seek to gather ample data ensuring high levels of prediction accuracy, otherwise an accurate descriptive identification of the system is unlikely (at least with FINE). Interestingly, in all the performance functions tested, regardless of their complexity, the operational accuracy of FINE is higher than the prediction accuracy of the initial prediction method. This fact implies that FINE, in addition to reconstructing the functional backbone, also acts as a smart predictor, which utilizes the assumption that the system in hand is functionally sparse to yield improved predictions of the behavior of the system in unknown states. Another interesting perspective on FINE's performance is given by the top summands detection rate measure; evidently, when the prediction accuracy rises above 75%, the top summands detection rate is higher than 80% even in the more complex cases ($m = 16$, $c = 6$). Compared with the performance of the FIN algorithm throughout our simulation experiments, FINE achieves better results in 96.4% of the cases, both in terms of descriptive accuracy (measured by the Jaccard coefficient) and dividends accuracy.

Figure 4 presents the prediction accuracy required for obtaining a desired level of descriptive accuracy, as a function of the complexity of the performance function. The results show a clear monotonic dependency of the required prediction accuracy on both the number and length of summands of the target functions. In biological applications, once a set of experiments has been performed, the prediction accuracy can be evaluated. Thereafter, by assuming the number and lengths of pathways taking place in the function studied, an estimate of the achievable CFN descriptive accuracy can be obtained. However, some caution is warranted since different predictors yield CFNs with different descriptive accuracies subject to the same initial prediction accuracy. Yet, the relative ordering between predictors is conserved across a span of prediction accuracy levels (data not shown).

FINE Analysis of the *Endo16 Cis*-regulatory System

To study the workings of FINE with biological data, we focus on the computational logic model constructed for the *cis*-regulatory system of the *endo16* gene of the sea urchin, *Strongylocentrotus purpuratus* presented by Yuh *et al.* (2001). This *cis*-regulatory system was studied thoroughly in a series of studies (*e.g.* Yuh *et al.*, (1996), 1998, 2001)). Combining the knowledge assembled by these studies allowed the formulation of a computational model (Yuh *et al.*, 2001) which describes in detail how the activity of the *endo16* gene is determined by it's *cis*-regulatory elements (transcription factor (TF) binding sites).

The main elements of the *endo16 cis*-regulatory system can be divided into three distinct groups. The two main groups, referred to as module A and B, correspond to two sets of TF binding sites, lying on two adjacent regions of the *cis*-regulatory apparatus. The elements in the third group correspond to whole clusters of binding sites (modules) lying upstream of modules A and B. Yuh *et al.* (2001) show how the elements in these groups interact to determine the expression level of the *endo16* gene throughout embryogenesis. Early in development, the *endo16* gene participates in the speci-

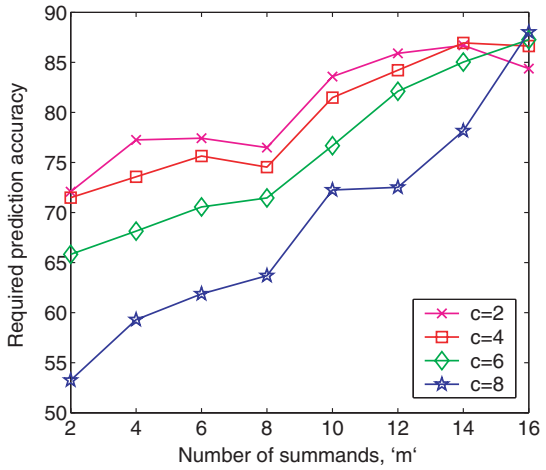


Fig. 4. *Performance of FINE on an ensemble of performance functions.* For each performance function studied we note the input prediction accuracy needed to achieve a desired descriptive accuracy—a combination of specificity above 55% and sensitivity above 75% (the y-axis). The figure depicts the dependency of the descriptive accuracy as a function of prediction accuracy over a range of different values of m, that is, over different number of summands in the performance function. Each line corresponds to a different value of c, that is, to a different maximum permitted length of summands in the performance function. For example, for a performance function with the parameters of ($m = 6$, $c = 4$), the prediction accuracy needed to achieve the descriptive accuracy criteria stated above is 70%. Observe, that the required prediction accuracy shows a monotonic increase both with c and m.

fication events that define the endomesoderm. This function is mainly dependent on module A. Later on, it serves as a gut-specific differentiation gene. This function is mainly dependent on module B. However, it still requires module A, whose main role at that point is to act as a mediator between module B and the basal transcription apparatus. In parallel, the upstream modules (referred to as modules F, E, D and C) mediated by a certain binding site in module A, were shown to serve as a repression subsystem whose role is to force spatial constraints over the activity of the *endo16*.

The computational model of Yuh *et al*. explains this spatial and temporal dependent activity as a direct result of the intactness of the elements in the three groups and of the concentration levels of the TFs which bind in them. Specifically, there are three TFs considered as *kinetic drivers*, in the sense that their activity profiles provide continuous, time-varying input to the system, whereas the rest of the TFs are perceived in only two discrete states—active or inactive.

In this study we aim to characterize the *endo16* regulation via FINE. We focus on a single, central, time point of 60 hours after fertilization, which is after the switch between module A and B took place (Yuh *et al*., 2005). The perturbation data is obtained by considering the effect of mutating or functionally knocking out different sets of binding sites. Our first task is to construct the full functional model and the Ground Truth CFN of the *endo16* system and use the resulting functional diagram to draw conclusions about the functional structure of the system. We then show how our observations match those given by Yuh *et al*. (2001) Our second task is to show how well can FINE reconstruct the GTCFN with limited training data. Finally, we compare the performance of FINE to that of the FIN algorithm.

We consider nine input variables, eight of them represent single binding sites—Otx, P and CG1 sites of module A, CY, CB1, CB2, UI and R sites of module B and a single variable Φ denoting the spatial repression subsystem, composed of the Z site of module A and the upstream modules F, E, D and C (Yuh *et al*., 1996, 1998). Each binding site variable can be assigned either with a value of '1' indicating that it is present and is occupied by its respective TF, or with a value of '0', indicating that it has been mutated or that its respective TF was inactivated or eliminated. The variable Φ is assigned with a value of '1' if and only if the repression system is inactive.

In order to accurately compute the dividend decomposition of the system, we need to obtain the activity levels of the *endo16* in response to all 2^9 perturbation configurations (Eq.(1)). We estimate the values of the basal promoter activity and of the concentration levels of the kinetic drivers (which bind at the UI, CB2 and Otx sites) at the time point of 60 hours after fertilization, reflecting their relative magnitudes in accordance with Yuh *et al*. (2005) (UI = 1, Otx = 0.2, CB2 = 0.3, Basal Activity = 0.2).

These values were then used to query the computational model for the corresponding activity levels. This provides the perturbation data needed for obtaining the full FIN functional model, via Eq.(2). In terms of our parameterization, the parameters of this functional model are ($n = 9$, $m = 7$, $c = 7$). We then apply FINE to obtain the GTCFN—a compact representation of the functional backbone of the *endo16 cis*-regulatory system. Two out of the seven summands were pruned during the GTCFN construction.

Figure 5 presents the functional diagram, obtained from the full functional model. Nodes 3 to 9 corresponds to different subsets of *cis*-regulatory elements. Node 2 correspond to the basal activity and node 1 is the output node. Each weighted edge corresponds to a dividend value. Dashed edges have zero weight and serve as Boolean *and* operators. The diagram shows a clear distinction between module *A* (nodes 3-4, squares), *B* (nodes 6-8, hexagons) and the hybrid subsystems (nodes 5, 9 ovals). The functional diagram of the GTCFN is, naturally, a subgraph of the full model's diagram, and can be obtained by excluding nodes 7 and 9 (indicated by a gray filling).

The functional diagram, based on the automated visualization module, clearly outlines the functional structure of the system and allows us to draw various insights regarding the different logical subsystems (or functional pathways) involved in determining the expression of the *endo16*. We point out a few such observations: (i) Node 5 acts as a bottle neck for the output of nodes 6-9. It depends on the sites P, CG1 and CB2. If one of these three elements is assigned with a value of zero, then the output of the system will depend solely on node 4 (the Otx site of module A). This subsystem is recognized by Yuh *et al*. as the *linkage subsystem*, which connects the output of module B into module A. (ii) The Otx site participates in two counteracting functional pathways, starting at nodes 9 and 4; If the pathway including nodes (9-6-5-3-1) is intact then the Otx site has no influence on the system, since its positive contribution via pathway (4-3-1) is totally repressed. This subsystem is recognized by Yuh *et al*. as the *BA intermodule input switch*, which represses the output of module A (via the Otx site) and leaves it solely as a mediator for the output of module B. (iii) The input elements in node 8 play only a single role in the system, which is to increase the output of node 6 by two fold. Yuh *et al*. term this as the *synergism*

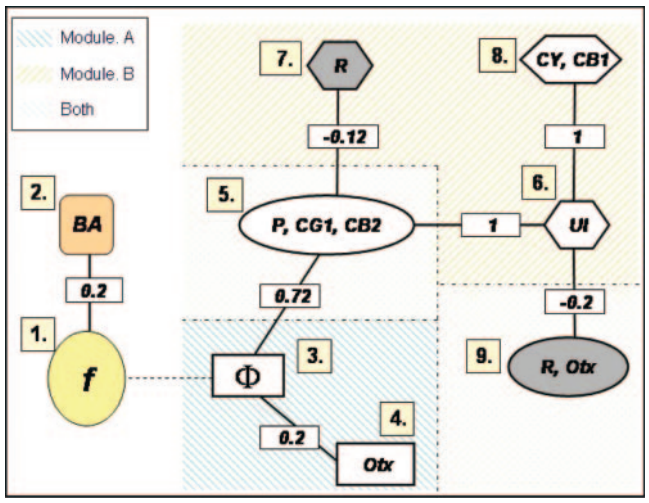


Fig. 5. *Functional diagram of the computational model of the* endo16 *cis-regulatory system.*

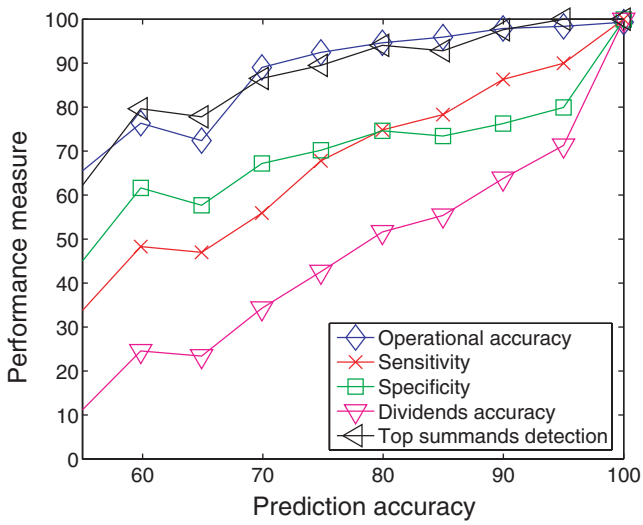


Fig. 6. *Performance of FINE analysis of the computational logic model of the* endo16 *cis-regulatory system.* The different FINE performance measures are plotted against the prediction accuracy of the data samples.

subsystem, which, mediated by the CY and CB1, steps up the output of UI. (iv) Taking a wider perspective, we see that module A is directly connected to the output node (assuming that the repression subsystem is inactive) whereas module B requires the intactness of module A in order to have an effect. On the other hand, the quantitative influence of module A alone (via node 4) is of low magnitude, compared to that of module B. This is in agreement with the fact that the role of module A at the time point we selected is mainly to serve as a mediator for the output of module B, whereas its own output is of less importance. Evidently, each of these observations, based on the functional diagram has an equivalence in the logical analysis of Yuh *et al*. This fact illustrates the utility of the FIN approach (and the FINE algorithm) as a tool for representation and analysis of biological systems.

The two summands which were pruned during the GTCFN construction correspond to the functional pathways connecting nodes 7 and 9 to the output node. The biological phenomena which correspond to these summands are both related to the R site: (i) The slight increase in the CB2 output which occurs once the R site is mutated (node 7). (ii) The BA intermodule input switch (node 9). Both these subsystems were recognized to have a marginal influence on the expression of the *endo16* at the time point examined (Yuh *et al*., 2001), and indeed, removing the two summands from the functional model reduces the operational accuracy by a mere 3.74%.

To study the relation between prediction accuracy and descriptive accuracy in the *endo16* system, we apply an assay similar to the one described in the simulated data section: We apply the FINE algorithm to a set of random samples of different sizes drawn out of the set of all 2^9 perturbation configurations. Figure 6 displays the performance of FINE as a function of the prediction accuracy yielded by the various data sets, in a manner analogous to that of Figure 3. Evidently, the results are quantitatively similar to the results on the simulated data (testifying that the simulated networks behave in a similar manner, CFN-wise, to the biological model).

The three most important summands in the GTCFN, according to the ranking scheme defined for the top summands detection rate measure, correspond to the pathways connecting the output node to nodes 2, 5 and 6 (pathways ($BA - f$), ($[P, CG1, CB2] - \phi - f$), ($UI - [P, CG1, CB2] - \phi - f$)). Indeed, the corresponding subgraph (induced by nodes 1, 2, 3, 5 and 6) includes the most influencing subsystems—the UI and CB2 sites where the two dominant kinetic drivers at the selected time point bind at, the *linkage subsystem*, which connects the output of module B into module A, and the basal activity which naturally effects the expression of the gene in all possible perturbation configurations and is therefore considered important as well. Interestingly, FINE identifies this subgraph, even when the prediction accuracy is poor. For example, with prediction accuracy of 60%, the top summands detection rate is 80%.

Comparing the performance of FINE to that of the FIN algorithm shows a clear superiority of the former. FINE achieves higher rates of both descriptive accuracy and dividends accuracy in 94.3% of our simulation experiments, producing significantly more compact and accurate models (data not shown).

CONCLUSIONS

This paper addresses a new challenge within the context of gene network analysis. In contrast to prevalent approaches, which aim to reveal the network of interactions between genes, our goal is to identify the underlying functional network. In this network description, the genes' states determine a quantitative phenotype of the network, and its architecture visualizes and explains how the studied function is actually carried out. To this end, we rigorously study the capabilities of FINE, a new, iterative algorithm based on the FIN algorithm presented in Kaufman *et al*. (2005). It is designed to handle sparse networks and leverages this assumption to achieve improved results. It is shown to successfully analyze simulated complex networks utilizing multi-knockout data, obtaining a simple and compact description of the underlying functional network. This compact representation delineates the important gene sets (pathways) and their functional influence. Our results demonstrate that in small-scale systems (of the scale of multi-knockouts currently

studied in biology), FINE successfully identifies the main subsets of genes with a low rate of false positives, obtaining a high success rate in identifying the top 3 important pathways in a system. Similar results are achieved in the biological example of the *endo16* regulation where FINE successfully extracts *in an automatic manner* the main known biological modules of the *cis*-regulatory network. The success rate in identifying the top three main functional modules in this system is above 80%, even with a predictive accuracy of 60%. Evidently, FINE outperforms the FIN when applied to sparse systems; this occurred in 96.4% of the simulated experiments we have conducted. However, in cases where the network architecture transpires such that two pathways completely cancel out the functional effect of each other in an almost precise manner, FIN might outperform FINE (such cases occurred in approximately 3% of our simulations). In addition, if the underlying system is not sparse and is composed of many functionally interacting pathways, FINE may lead to an erroneous, grossly over-pruned network, while FIN is likely to lead to a significantly more accurate description.

Applying the FINE visualization module in the analysis of the biological example, demonstrates the correspondence between insights that can be gained via the functional diagram and the pertaining biological knowledge of the *endo16* regulation, summarized in its Boolean logic description.

The second main theme addressed in this study is the question of how many experiments are needed to successfully identify the CFN. Our results show that the descriptive and operational accuracy of the CFN are dependent on the prediction accuracy of the experimental set in hand. However, the complexity of the underlying network further modulates the required prediction accuracy in a significant manner. On a more quantitative level, the simulated data results provide the biologist with general ballpark numbers as to what levels of prediction accuracy he must achieve to obtain a desired level of descriptive accuracy. To this end, however, the biologist must have some a-priori gross estimation of the complexity/architecture of the system in hand.

FINE is not limited to small-scale systems, and is potentially scalable to much larger networks, under some constraints; We are now developing and studying a *k-bounded* variant of the FINE algorithm, in which we assume that, even if the system is large, only a bounded set of k elements significantly influences the investigated function (and different tasks in the system may be realized by different, possibly overlapping bounded sets). Other directions for future work include applying the FINE to multiple functions concomitantly—in such cases one can identify and classify functions according to their functional backbones or extract common features within the obtained functional backbone. Importantly, if the multiple functions under investigation are assumed to share similar functional modules the construction of the CFN can benefit from a higher degree of accuracy by validating the importance of the chosen dividends across the different functions.

Since FINE is model independent, it is potentially applicable to a wide variety of systems. The only requirement is that the function performed by the system can be measured under different discrete states of its elements e.g. perturbed, silenced, inactive, enhanced, over-expressed and so on. Notably, the 'elements' perturbed need not necessarily be single elements and can be sub-modules of a system. For example, in the sea urchin model, we can relate to the *endo16* CFN as a single node in a more comprehensive developmental network, leading to a hierarchical functional view of the system. It is likely that the most immediate and rewarding current application of FINE is in the analysis of multi-perturbation studies in genetics, in view of the rapid recent advances in gene silencing with RNAi. The FINE algorithm offers a viable way for the accurate identification of the main functional pathways in biological systems.

ACKNOWLEDGEMENTS

N.Y. is supported by the Tel-Aviv university president and rector scholarship, A.K. is supported by the Yeshaya Horowitz Association through the Center of Complexity Science. E.R's research is supported by grants from the Israeli Science Foundation (ISF), the Yeshaya Horowitz Center of Complexity Science, and from the Tauber Fund.

REFERENCES

Carpenter,A.E. and Sabatini,D.M. (2004) Systematic genome-wide screens of gene function. *Nat. Rev. Genet.*, **5**, 11–22.

Barabasi,A.L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, **5**, 101–113.

Davidson,E.H. *et al.* (2002) A genomic regulatory network for development. *Science*, **295**, 1669–1678.

Gu,Z. *et al.* (2003) Role of duplicate genes in genetic robustness against null mutations. *Nature*, **421**, 63–66.

Kaufman,A. *et al.* (2005) Quantitative analysis of genetic and neural multi-perturbation experiments. *PLoS Comput. Biol.*, **1**(6): e64.

Tong,A.H. (2004) Global mapping of the yeast genetic interaction network. *Science*, **303**, 808–813.

Yuh,C.H., Bolouri,H. and Davidson,E.H. (2001) cis-regulatory logic in the endo16 gene: switching from a specification to a differentiation mode of control. *Development*, **128**, 617–629.

Kaufman,A., Kupiec,M. and Ruppin,E. (2004) Multi–knockout genetic network analysis: The Rad6 example. *Proceedings of IEEE Computational Systems Bioinformatics Conference (CSB'04)*, pp. 332–340.

Hammond,S.M., Caudy,A.A. and Hannon,G.J. (2001) Post-transcriptional gene silencing by double-stranded RNA. *Nature Rev. Gen.*, **2**, 110–119.

Ideker,T.E., Thorsson,V. and Karp,R.M. (2000) Discovery of regulatory interactions through perturbation: inference and experimental design. In *Proc. of the Pac Symp. on Biocomputing*, pp. 305–316.

Pe'er,D. *et al.* (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics* 17 Suppl 1:S215–S224.

Ideker,T.E. *et al.* (2001) Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, **292**, 929–934.

Tegner,J. *et al.* (2003) Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proc. Natl. Acad. Sci. USA*, **100**, 5944–5949.

Keinan,A. *et al.* (2004) Fair attribution of functional contribution in artificial and biological networks. *Neural Computation*, **16**, 1887–1915.

Thieffry,D. *et al.* (1998) From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in escherichia coli. *BioEssays*, **20**, 433–440.

Jeong,H. *et al.* (2000) The large-scale organization of metabolic networks. *Nature*, **407**, 651–654.

Grabisch,M., Marichal,J.L. and Roubens,M. (2000) Equivalent representations of a set function with applications to game theory and multicriteria decision making. *Mathematics of Operations Research*, **25**, 157–178.

Businger,P.A. and Golub,G.H. (1965) Linear least squares solution by householder transformation. *Numer. Math.*, **7**, 269–276.

Yuh,C.H. and Davidson,E.H. (1996) Modular *cis*-regulatory organization of endo16, a gut-specific gene of the sea urchin embryo. *Development*, **122**, 1069–1082.

Yuh,C.H., Moore,J.G. and Davidson,E.H. (1996) Quantitative functional interrelations within the *cis*-regulatory system of the s. purpuratus endo16 gene. *Development*, **122**, 4045–4056.

Yuh,C.H., Dorman,E.R. and Davidson,E.H. (2005) Brn1/2/4, the predicted midgut regulator of the endo16 gene of the sea urchin embryo. *Dev. Biol.*, **281**, 286–298.

Yuh,C.H., Bolouri,H. and Davidson,E.H. (1998) Genomic *cis*-regulatory logic: Functional analysis and computational model of a sea urchin gene control system. *Science*, **279**, 1896–1902.