Genomics 102 (2013) 202–208

Contents lists available at ScienceDirect

Genomics

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

Exploitation of genetic interaction network topology for the prediction of epistatic behavior

Gregorio Alanis-Lobato [1], Carlo Vittorio Cannistraci [1], Timothy Ravasi *

Integrative Systems Biology Lab, Biological and Environmental Sciences and Engineering Division, Computer, Electrical and Mathematical Sciences and Engineering Division, Computational Bioscience Research Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Kingdom of Saudi Arabia
Division of Medical Genetics, Department of Medicine, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093 USA

ARTICLE INFO

Article history:
Received 25 June 2012
Accepted 17 July 2013
Available online 25 July 2013

Keywords:
Genetic epistasis
Gene networks
Projections and predictions
Systems biology
Computational biology

ABSTRACT

Genetic interaction (GI) detection impacts the understanding of human disease and the ability to design personalized treatment. The mapping of every GI in most organisms is far from complete due to the combinatorial amount of gene deletions and knockdowns required. Computational techniques to predict new interactions based only on network topology have been developed in network science but never applied to GI networks.
We show that topological prediction of GIs is possible with high precision and propose a graph dissimilarity index that is able to provide robust prediction in both dense and sparse networks.
Computational prediction of GIs is a strong tool to aid high-throughput GI determination. The dissimilarity index we propose in this article is able to attain precise predictions that reduce the universe of candidate GIs to test in the lab.

1. Introduction

A genetic interaction (GI) or *epistasis* is detected if a double mutation generates a deviation in phenotype that is worse (negative interaction) or better (positive interaction) than that generated by the combination of the single mutant phenotypes [1]. GIs play an important role in untangling the relationships between genotype and phenotype, advance our understanding of human disease [2–4] and improve our ability to design personalized treatment plans. Nevertheless, mapping every GI is far from complete, even in model organisms. Here is where computational prediction of novel GIs comes into play.

Computational techniques to predict new GIs aim to reduce costs incurred by their experimental detection and target good candidates to test in the lab. Current attempts to computationally predict GIs depend on biological information that, for some organisms, might not be available. Examples of these efforts are the integration of data that characterize *epistasis*, such as expression, physical interaction, or functional annotations to train probabilistic decision trees [5] or to apply logistic regression [6]. Other endeavors involve the overlap of data coming from different networks (Protein Interaction Networks, Gene Ontology Networks, Co-expression Networks, etc.) and the application of random walks [7] or an ensemble of classifiers [8].

We propose the exploitation of the biological information stored exclusively in the network topology that should be shaped by the genomic properties characterizing the organism under investigation. To this effect, parameter-less neighborhood-based (both general-purpose and bio-inspired) and network-embedding techniques are applied to the GI networks (GINs) of two different organisms (worm and yeast) the first of which is sparser than the second one. The reliability of these techniques and the impact of sparse network architecture on their prediction performance are analyzed and discussed. We also propose a graph dissimilarity index that proves better performance in candidate GI prediction, for the networks here considered.

2. Data and algorithms

2.1. Datasets

This work focuses on negative interactions due to their known impact on essential biological functions [9]. The datasets used correspond to GIs in *Saccharomyces cerevisiae* (yeast) and *Caenorhabditis elegans* (worm), detected by [10,11] respectively and downloaded from BioGRID 3.1.85 [12].

Self-interactions and redundant links were removed from these datasets to constitute a Worm GIN of 457 nodes and 1242 links (average node degree = 5.44) and a Yeast GIN of 3842 nodes and 52179 links (average node degree = 27.16). Notice the latter is ~5 times denser than the former.

* Corresponding author at: Division of Biological and Environmental Sciences and Engineering, Division of Computer, Electrical and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology, Thuwal 23955-6900, Kingdom of Saudi Arabia.
E-mail address: timothy.ravasi@kaust.edu.sa (T. Ravasi).
URL: http://systemsbiology.kaust.edu.sa (T. Ravasi).
[1] The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

0888-7543/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ygeno.2013.07.010

2.2. Neighborhood-based prediction

General-purpose neighborhood-based techniques have been created for link prediction in different kinds of networks: social, roadmaps, citation, collaboration, etc. [13]. Bio-inspired techniques were created to either assess reliability of interactions in biological networks [14] or predict protein function [15]. Later, these techniques were applied to protein interaction prediction [16,17]. Both approaches rely on the number of neighbors that two non-directly connected nodes have and assign a likelihood score to this pair of nodes based on the equations listed in Table 1.

The simplest techniques are Jaccard (JC) [18], Common Neighbors (CN) and Preferential Attachment (PA) [19]. JC assigns higher likelihood scores as the set of common interactors that coincides with the set of all available neighbors and CN does the same for pairs of nodes that share many interactors (see Eqs. (1) and (2)). PA, on the other hand, gives high scores when both nodes have a large number of neighbors: if one of the nodes has a low number of interactors, the score is reduced (see Eq. (3)). In contrast, Adamic & Adar (AA) [20] and Resource Allocation (RA) [21] are two similar indices that give more importance to CNs with low degree (see Eqs. (4) and (5)).

Interaction Generality (IG1) [22] is based on the fact that partners of sticky proteins and self activators do not interact with anything else in the network (see Eq. (6)). Czekanowski–Dice Dissimilarity (CDD) [23] and Functional Similarity Weight (FSW) [15] have their basis in direct and indirect functional association: the more common neighbors the two nodes have, the more they are likely to share function or be involved in the same processes (see Eqs. (7) and (8)). CDD is a dissimilarity index, meaning that if two nodes are similar, CDD is close to zero. On the contrary, FSW is a probability and the closer it is to 1, the more likely it is that two nodes interact. Several other bio-inspired techniques have been proposed but they are computationally expensive [24] and both FSW and CDD have proven to be the best options [14,16]. We took IG1 into account because it is considered a pioneering technique in the bio-inspired category.

2.3. Network embedding prediction

Network embedding prediction is based on the idea that the network topology is shaped in a space of high dimensions and that, once its components are embedded into a reduced space, interacting nodes are mapped close to each other [25]. Thus, nodes that are not connected in the original network but that are close to each other in the low-dimensional space are likely to interact.

This type of prediction consists of choosing a graph metric and computing all-pair distances between nodes in the network to generate a distance matrix D. The matrix is mapped to a reduced space by extraction of its highest eigenvalues and eigenvectors, obtained via singular value decomposition (SVD), which are used to recover the coordinates of the nodes in the low-dimensional space. We note that SVD can be applied to D or to its centered version. In the first case we obtain a non-centered embedding and the algorithm's time complexity is: $O(N^2)$, where N is the number of network nodes; in the second case we obtain a centered embedding and the time complexity is the same as for matrix centering: $O(N^3)$. Then, a node-proximity estimation is computed in the reduced space and the distances between non-directly connected nodes correspond to their assigned likelihood scores.

Two network embedding techniques are applied in this work: Isomap (ISO) [16,26] and Minimum Curvilinear Embedding (MCE) [27]. ISO computes all-pair shortest-path distances to generate D, whereas MCE extracts the Minimum Spanning Tree (MST) out of the network and then computes all-pair shortest-path lengths over it. At this point, SVD can be applied to the non-centered D (ncISO and ncMCE) or to its centered version (cISO and cMCE). Once D is mapped to the space of low dimensions, and the coordinates of the nodes are recovered, the network is reconstructed and the proximity estimation of choice (in this case the shortest-path over the network) is computed to assign scores [28]. We clarify that the network embedded in the reduced space acquires weights for its edges. These weights correspond to the Euclidean distance between the respective connected nodes.

To determine the best dimension to embed the network into, we adopted two approaches recently proposed by Cannistraci and colleagues [28]: dimension determination by Area Under the Receiver Operating Characteristic (ROC) Curve (AUC, Fig. 1A) and by Resolution (Res, Fig. 1B).

In the AUC approach (based on the work of You et al. [16] in 2010), the network is embedded into dimension 1 and the distances between all nodes are computed in the reduced space. These distances are sorted by increasing length and a threshold ε is varied from 0 to the longest distance to quantify the number of True Positives (links from the original network that pass the ε cut), False Negatives (links from the original network that do not pass the ε cut), True Negatives (non-directly connected nodes that do not pass the ε cut) and False Positives (non-directly connected nodes that pass the ε cut). With these numbers at each ε, we can compute the True Positive Rates and False Positive Rates and generate a ROC curve. The process is repeated for higher dimensions, until the difference between the AUCs for dimension *dim* and dimension *dim*-1 is less than 0.001 (resulting in *dim* as the selected dimension, see Fig. 1A). In several tests we found that 0.001 is sufficiently small to suggest that no better performance would be achieved if nodes were mapped to higher dimensions, for this reason we chose it as a stopping flag in the AUC criterion. The problem with this approach is that it takes the original network as ground truth, which may not be accurate given the amount of non-real interactions included in the topology due to experimental bias or defects [29,30].

The Res approach addresses the above-mentioned issue. The best dimension according to this criterion is the one that provides good discrimination between good and bad candidates for interaction, i.e. the more different the distances in the reduced space, the higher the resolution and the better the dimension. Application of Res requires the use of Eq. (9). If $scores_{dim}$ is the set of scores assigned to the candidate interactions in dimension *dim*, $unique(scores_{dim})$ is a function that discards the duplicates in the set and returns only its distinct elements. Later the standard deviation σ of the unique scores is computed to assess the quality of the resolution provided to finally divide the result by *dim* to penalize high dimensions, which have been shown not to

Table 1
Table of equations for the neighborhood-based techniques. Techniques marked with a G correspond to general-purpose neighborhood-based approaches, while techniques marked with a B correspond to bio-inspired approaches. x and y are network nodes; $\Gamma(x)$ refers to the set of neighbors of x, whereas $\gamma(x)$ refers to the set of neighbors of x including x; $|\Gamma(x)|$ refers to the cardinality of set $\Gamma(x)$ and Δ is the symmetric set difference. The numbers in parentheses identify each equation throughout the text.

Technique	Equation	
(G) Common neighbors	$\lvert\Gamma(x)\cap\Gamma(y)\rvert$	(1)
(G) Jaccard	$\lvert\Gamma(x)\cap\Gamma(y)\rvert/\lvert\Gamma(x)\cup\Gamma(y)\rvert$	(2)
(G) Preferential attachment	$\lvert\Gamma(x)\rvert\cdot\lvert\Gamma(y)\rvert$	(3)
(G) Resource allocation	$\sum_{s\in\Gamma(x)\cap\Gamma(y)} 1/\lvert\Gamma(s)\rvert$	(4)
(G) Adamic & Adar	$\sum_{s\in\Gamma(x)\cap\Gamma(y)} 1/\log(\lvert\Gamma(s)\rvert)$	(5)
(B) IG1	1 plus the number of nodes that directly interact with x or y and nothing else in the network.	(6)
(B) CD-Dist	$\lvert\gamma(x)\,\Delta\,\gamma(y)\rvert$ / ($\lvert\gamma(x)\cup\gamma(y)\rvert$ + $\lvert\gamma(x)\cap\gamma(y)\rvert$)	(7)
(B) FSWeight	$(2\lvert\gamma(x)\cap\gamma(y)\rvert$ / ($\lvert\gamma(x)-\gamma(y)\rvert$ + $2\lvert\gamma(x)\cap\gamma(y)\rvert$ + λ_x)) × ($2\lvert\gamma(x)\cap\gamma(y)\rvert$ / ($\lvert\gamma(y)-\gamma(x)\rvert$ + $2\lvert\gamma(x)\cap\gamma(y)\rvert$ + λ_y)) where $\lambda_x = \max(0, n_{avg} - \lvert\gamma(x)\rvert)$ and n_{avg} is the average node degree of the network	(8)