

1 **Solving the influence maximization problem reveals regulatory**  
2 **organization of the yeast cell cycle.**

3

4 ***Short title: Influence maximization on cell cycle regulatory networks.***

5

6 David L Gibbs<sup>1</sup>, Ilya Shmulevich<sup>1\*</sup>

7

8 <sup>1</sup> Institute for Systems Biology, Seattle, Washington

9

10 \*Corresponding author: David L Gibbs<sup>1</sup> ([david.gibbs@systemsbiology.org](mailto:david.gibbs@systemsbiology.org))

11

12 **Keywords:** Networks, Information theory, Gene regulatory networks, Gene expression,  
13 Computational Models

14

15 **Abstract**

16           The Influence Maximization Problem (IMP) aims to discover the set of nodes with  
17 the greatest influence on network dynamics. The problem has previously been applied  
18 in epidemiology and social network analysis. Here, we demonstrate the application to  
19 cell cycle regulatory network analysis of *Saccharomyces cerevisiae*.

20           Fundamentally, gene regulation is linked to the flow of information. Therefore, our  
21 implementation of the IMP was framed as an information theoretic problem on a  
22 diffusion network. Utilizing all regulatory edges from YeastMine, gene expression  
23 dynamics were encoded as edge weights using a variant of time lagged transfer  
24 entropy, a method for quantifying information transfer between variables. Influence, for  
25 a particular number of sources, was measured using a diffusion model based on  
26 Markov chains with absorbing states. By maximizing over different numbers of sources,  
27 an influence ranking on genes was produced.

28           The influence ranking was compared to other metrics of network centrality.  
29 Although ‘top genes’ from each centrality ranking contained well-known cell cycle  
30 regulators, there was little agreement and no clear winner. However, it was found that  
31 influential genes tend to directly regulate or sit upstream of genes ranked by other  
32 centrality measures. This is quantified by computing node reachability between gene  
33 sets; on average, 59% of central genes can be reached when starting from the  
34 influential set, compared to 7% of influential genes when starting at another centrality  
35 measure.

36           The influential nodes act as critical sources of information flow, potentially having  
37 a large impact on the state of the network. Biological events that affect influential nodes

38 and thereby affect information flow could have a strong effect on network dynamics,  
39 potentially leading to disease.

40 Code and example data can be found at: <https://github.com/Gibbsdavidl/miergolf>

#### 41 ***Author Summary***

42 The Influence Maximization Problem (IMP) is general and is applied in fields such as  
43 epidemiology, social network analysis, and as shown here, biological network analysis.

44 The aim is to discover the set of regulatory genes with the greatest influence in the  
45 network dynamics. As gene regulation, fundamentally, is about the flow of information,  
46 the IMP was framed as an information theoretic problem. Dynamics were encoded as  
47 edge weights using time lagged transfer entropy, a quantity that defines information  
48 transfer across variables. The information flow was accomplished using a diffusion  
49 model based on Markov chains with absorbing states. Ant optimization was applied to  
50 solve the subset selection problem, recovering the most influential nodes. The influential  
51 nodes act as critical sources of information flow, potentially affecting the network state.

52 Biological events that impact the influential nodes and thereby affecting normal  
53 information flow, could have a strong effect on the network, potentially leading to  
54 disease.

#### 55 ***Introduction***

56 Living systems dynamically process information. Cell surface receptors capture  
57 information from the environment and relay the information by internal signaling  
58 pathways [1,2]. Information is transferred, stored, and processed in the cell via  
59 molecular mechanisms, often triggering a response in the regulatory program. These  
60 types of dynamic genetic regulatory processes can be modeled with network

61 information flow analysis.

62 Network flows embody a general class of problems where some quantity flows  
63 from source nodes, across the edges of a graph, draining in sink nodes. Various forms  
64 of network flow methodologies have found success in algorithms such as Hotnet,  
65 ResponseNet, resistor networks, and others [3,4,5].

66 Recently, the influence maximization problem (IMP) has received a great deal of  
67 interest in social network analysis and epidemiology as a general method for  
68 determining the relative importance of nodes in a dynamic process [6,7]. The IMP aims  
69 to discover a set of source nodes that, after applying a diffusion model, covers as much  
70 of the network as possible [8,9]. Examples are found in modeling the spread of  
71 infectious disease in social networks and in identifying optimal targets for vaccination  
72 [10]. Propagation of infection does not follow algorithmically defined paths on graphs,  
73 i.e. shortest paths, but instead flows on all possible paths. Similar to quantities of virus,  
74 information can also be treated as a quantity flowing on networks [11,12,13].

75 A variant of ant optimization was used to solve the IMP. Although, ant  
76 optimization is best known in path optimization, it can also be applied to subset  
77 selection problems [14,15,16]. In ant optimization, ants construct potential solutions, as  
78 sets, which are scored and reinforced, encouraging good solutions in later iterations. In  
79 this work, the result of the optimization procedure is an optimal, or nearly optimal, set of  
80 nodes that maximizes network cover when applying the diffusion algorithm developed  
81 by Stojmirović and Yu [12]. In application to biological networks, the IMP essentially  
82 remains an unexplored area of research [17]. We then used the IMP to study the  
83 regulatory structure underlying the yeast cell cycle.

84 To understand topologically where the influential genes are situated, we compare  
85 the IMP solution sets to gene sets derived from other centrality metrics, such as degree  
86 centrality [18], betweenness-centrality where shortest paths are considered [19], and  
87 PageRank, where only incoming flows are considered [20].

88 The cell cycle process in *Saccharomyces cerevisiae* is well studied, but is not  
89 completely characterized [21]. Regardless, it is apparent that cell cycle regulation is  
90 controlled by a network that dynamically processes signals. From the bench, we have  
91 limited ways of observing the process, such as using gene expression data and  
92 cataloging the patterns of periodicity. To gain further understanding of the regulatory  
93 structure, we performed used time series data and publicly available regulatory  
94 databases to solve the IMP (Fig 1) [22,23].

95 **Figure 1.) Analysis workflow.** All regulatory edges from the YeastMine DB  
96 formed the regulatory network scaffold. Using time series gene expression data, time  
97 lagged transfer entropy was calculated and each edge was tested using a permutation-  
98 testing framework. The resulting network was used for solving the Influence  
99 Maximization Problem.

## 100 **Results**

### 101 **Filtering regulatory edges using time lagged transfer entropy**

102 Statistical metrics such as Pearson correlation are sometimes used to estimate  
103 the activity of regulatory edges. However, processes in biology do not instantaneously  
104 complete, and so various time lags are introduced to account for propagation time (SI  
105 Fig 1) [24]. Additionally, genetic regulatory interactions are directional; transcription  
106 factors act on genes, and not the other way around. So, although Pearson correlation is

107 simple, there are more appropriate metrics to use with time series data, such as transfer  
108 entropy. Transfer entropy (TE) is a model-free method that attempts to quantify  
109 information transfer between two variables in a directional manner. Permutation-based  
110 statistics can be applied to assess the significance of TE. TE was computed over  
111 multiple time lags and summed. This avoids choosing a single time lag and is termed  
112 'sum-lagged transfer entropy' or SLTE [35].

113 Using time series data for 5,080 measured genes and 26,827 genetic regulatory  
114 edges from YeastMine, both time-lagged Pearson correlation and sum-lagged transfer  
115 entropy was computed for all regulatory edges. Edges were removed if empirical p-  
116 values were greater than  $1/p_n$  where  $p_n$  is the number of permutations ( $p_n = 50,000$ ).

117 Time lagged Pearson correlation, where the maximum correlation is returned  
118 after considering a range of time lags (0-6 time steps), resulted in 7,729 edges,  
119 containing 3,216 nodes. Significant edge weights had a median correlation of 0.67.  
120 Most of the edges (76%) showed a maximum correlation when using a time-lag of zero.

121 The metric of interest, SLTE, resulted in 1,987 significant edges containing 1,147  
122 nodes with median weights of 1.37 (Fig 2).

123 **Figure 2.) The resulting BioFabric network after significance testing.** Nodes  
124 are shown as horizontal lines, with edges shown as vertical lines connecting nodes.  
125 High degree nodes can be seen as 'wedges' in the graph.

126 The overlap between the Pearson correlation and SLTE networks is moderate;  
127 only 12.8% of the edges in the correlation network are shared with the SLTE network  
128 (986 of 1,987 edges in the SLTE network or 49.6%), and while all SLTE nodes are  
129 found in the correlation network, only 33.9% of the correlation nodes are found in the

130 SLTE network. When comparing the two different edge weights, Pearson's and SLTE,  
131 on matched edges, the Pearson's correlation between edge weights was low (0.39).  
132 Additionally, the mean node degree distribution in the correlation network is much  
133 higher than that of the SLTE network. For example, the SFP1 gene has degree 589 in  
134 the correlation network, compared to 60 in the SLTE network, summing both in- and  
135 out-edges. The high node degree in the correlation network suggests that correlation  
136 testing may be overly permissive, with less informative edge weights.

137        Clauset, Shalizi, and Newman's method for statistically determining whether a  
138 network is 'scale-free' showed that the SLTE network is not [25]. On the SLTE network,  
139 the result showed  $\alpha = 2.18$ , which is concordant with power law networks. However,  
140 the goodness of fit test using the Kolmogorov-Smirnov statistic returned a p-value of  
141 0.03, indicating that only a small fraction of the simulated scale-free distributions are  
142 "close" to the observed degree distribution.

143        In the rest of the analysis, only the transfer entropy network is used, since it is  
144 clear that the correlation-based network is not a super-set of the transfer entropy  
145 network, does not agree in the weighting, and is likely overly permissive with regard to  
146 active interactions.

#### 147 **Iteratively solving the influence maximization problem provides a ranking**

148        Using transfer entropy to quantify information flow, if an upstream node transfers  
149 information to a downstream node, respecting edge directions, the downstream node is  
150 said to be 'influenced'. The maximization problem is to find a set of nodes, that when  
151 treated as information sources, influence the largest proportion of the network.

152        The Influence Maximization Problem (IMP) was solved over a range of values for

153  $K$ , the number of source nodes. The influence score, representing a network cover,  
154 increased quickly for small values of  $K$ , gradually leveling out. Setting  $K=45$  source  
155 nodes (3.9% of the network) produced a maximum network cover of 1064 nodes  
156 (92.8%). Beyond  $K=45$ , the score became saturated (see SI Fig 2).

157 As the algorithm is stochastic, the range of  $K$  (from 1 to 45) was run twice and an  
158 average count was made on the number of times genes were selected. The ranking  
159 produced by each run was highly stable, eliminating the need for a large number of  
160 runs. The top ranked gene FKH1, was selected on average 44 times, followed by two  
161 genes, GCN5 and RFX1, that were selected on average 43 times. Overall, 49 genes  
162 were selected in at least one run.

### 163 **Comparing influence to traditional metrics of centrality reveals similarities**

164 To provide a basis for comparison to the ranked influencers, 15 different  
165 centrality measures were computed on the SLTE network. The list of centrality metrics  
166 can be found in Table 1 along with a brief description. Further description of these  
167 metrics can be found in supplementary text. The top 20 influencers and associated  
168 metrics are found in Table S2.

169

170 **Table 1. Description of centrality metrics**

Tool	Centrality	Notes
igraph	Alpha Centrality	Generalization of eigen-centrality for weighted directed graphs.
	Articulation	Also called cut vertices, removal creates separate components.
	Authority	Kleinberg's centrality scores, based on principal eigenvector
	Betweenness	Uses directed and weighted edges to sum shortest paths through a vector.
	Closeness	Inverse of the average length of shortest paths to a vertex in the graph.

---

	Degree	Sum of edges for each vertex.
	Ego, 1 step	Size of the neighborhood one step out.
	Ego, 2 steps	Size of the neighborhood two steps out.
	Eigen Centrality	Eigenvector centrality on the undirected graph using weights.
	Page Rank	Google Page Rank, using directed and weighted edges.
	Power Centrality	Boncich power centrality. Accounts for connectivity of neighbors.
	Strength	Sum of edge weights.
	Unconstraint	Equal to (1 – Constraint), using Burt’s constraint method
	SubGraph Centrality	Participation of each node in all subgraphs
R	SVD	Right principal eigenvector using SVD, directed graph using weights.
python/ scipy	Influence Ranking	Ranked using influence maximization algorithm as described.

---

171

172 Each measure of centrality imparts a ranking over genes in the graph. The top  
173 2.5% of genes was selected from each metric, providing approximately 30 genes for  
174 each measure. In metrics with binary values, such as articulation, everything greater  
175 than zero was selected. A Jaccard index was computed for each pair of centrality  
176 measures (Fig 3). Although some clustering is observed among centrality metrics,  
177 especially for node-degree related measures, there remains substantial disagreement in  
178 top ranked genes.

179 **Figure 3.) The Jaccard index was used to compare centrality measures.** The  
180 top 2.5% of ranked genes from 16 different centrality measures were compared using  
181 the Jaccard index, which gives values of 1.0 for perfect agreement between sets, and 0  
182 for disjoint sets. There were approximately 30 genes in each set. The dendrogram

183 shows clustering among measures.

184           The top ranked influential genes are not found among highly ranked genes in  
185 eigenvector based centrality measures including authority, alpha centrality, and the SVD  
186 derived eigenvector. However, eigenvector measures of centrality contain important  
187 genes that are not found in other lists. For example, CLN1 was selected by the SVD  
188 derived eigenvector while it was not found on the betweenness list. Overall, no ranked  
189 list contained a definitive set of cell cycle related regulators. Across measures, gene set  
190 enrichment showed a wide variety of associations with biological processes, illustrating  
191 differences in the gene rankings.

192           **Figure 4.) Highly influential genes tend to be selected by other centrality**  
193 **metrics.** Genes are sorted by influence ranking in rows (top to bottom), and centrality  
194 metrics are found in columns. Genes in orange were influence ranked, but not selected  
195 as being in the top 2.5%.

196

#### 197 **Influential topology in the regulatory network**

198           We have found that within the regulatory network structure, the influential genes  
199 tend to be situated upstream of genes selected by other centrality measures (Fig 5).

200           **Figure 5.) Topology of influential nodes.** Highly influential nodes (blue) tend to  
201 be upstream of other genes (red) selected by a variety of centrality metrics. Overlapping  
202 genes are shown in purple.

203

204           For example, the influencer genes act as regulators for genes selected by alpha  
205 centrality, while no genes selected by alpha centrality regulate the influencer genes.

206 The same is found for the SVD-derived-eigencentralities and betweenness sets. In some  
207 cases, there is a fair amount of overlap in the top-level regulators, such as among the  
208 high degree nodes. But, overall, we see the influencers stay as top-level regulators to  
209 genes selected by other centrality measures. This can be quantified by computing the  
210 fraction of reachable genes, starting at a given measure, excluding overlapping genes  
211 (Fig 6). For example, starting at the set of influential genes, 80% of the betweenness  
212 selected genes can be reached, while starting at the betweenness genes, only 10% of  
213 influencers can be reached. Starting at the influencer genes, 41% of degree central  
214 nodes can be reached, while only 10% of influencers can be reached from the degree  
215 central nodes. Starting from every centrality measure, the fraction of reachable nodes is  
216 less compared to starting from the influential genes. On average, 59% of “central  
217 genes” can be reached when starting at the “influential set”, compared to 7% of  
218 reachable influential genes, after starting from another measure. The influencer genes  
219 are topologically central and connect important genes found by other centrality  
220 measures.

221 **Figure 6.** Influence can be quantified by computing node reachability. In (A), an  
222 example of node reachability is shown. After starting from a defined set of nodes,  $O$ , a  
223 node,  $v$ , is considered reachable if there exists a directed edge leading from  $O$  to  $v$ . For  
224 example, starting at the set of influential nodes, 80% of top ranking nodes using the  
225 betweenness measure can be reached, compared to only 10% of influential nodes after  
226 starting at the “betweenness nodes”. In (B) node reachability over all centrality  
227 measures is aggregated in a boxplot.

228 In Eser et al., 32 hypothesized cell cycle regulators were named [23]. Comparing

229 the top ranked influential genes, we see again that the influential genes are immediately  
230 upstream of the Eser TFs (Fig 7). Although SWI6 was not selected in the top 2.5% of  
231 influencers, it is found as a ranked influencer overall. BAS1, on the other hand was not  
232 a ranked influencer, and interestingly was not ranked by any other metric of centrality  
233 although it has been associated with cell cycle regulation.

234 **Figure 7.) Influence ranking of cell cycle related transcription factors.** Of the  
235 32 cell cycle related transcription factors given by Eser et al. (red), most are directly  
236 downstream of influential genes (blue). Purple shows an overlap between influential and  
237 Eser selected genes.

238 Recently a computational cell cycle model that successfully accounts for 257 of  
239 263 phenotypes [26] was published. In total, 29 genes were extracted from the model  
240 where genes making protein complexes were considered separately (SWI6 and SWI4  
241 were used instead of SBF). The full YeastMine network scaffold contained 28 of the 29  
242 genes (CDC55 was not present), and 15 genes were in the SLTE network. Only three  
243 genes were ranked as influencers (MBP1, SWI4 and SWI6).

244 The model is composed of seven modules, each containing between two to eight  
245 genes. The MEN module, containing only two genes, was the only module that did not  
246 contain any genes from the SLTE network. Among the other six modules, each  
247 contained between one to five SLTE genes. While most of the Tyson model genes are  
248 not ranked influencers, they are immediately regulated by influential genes. SWE1 is  
249 regulated by 8 ranked genes. CDC20 is regulated by 4 ranked genes. CLB5 is regulated  
250 by 7 ranked genes. SIC1 is regulated by 5 ranked genes. So in almost all cases, the  
251 Tyson model genes are not regulated by a single influencer, but by multiple influencers.

252

253 ***Discussion***

254           Transfer entropy has been shown to be useful in quantifying information transfer.

255       Here, we show that computing edge weights by summing over time lags, and using a

256       permutation testing framework, leads to biologically salient network structures. Even

257       though the network was constructed by considering all possible regulatory edges, it

258       recovers much of the structure and functional enrichment that one would expect, as

259       demonstrated by the lists of genes returned by commonly used centrality metrics like

260       betweenness and degree.

261           Since the edges encode dynamics of gene expression by representing

262       information transfer between regulators and targets, flow based methods are particularly

263       relevant. In network flows, some imaginary quantity 'flows' from node to node, limited by

264       the capacity of the edges. Here the capacity is represented by the sum of transfer

265       entropies over a small number of time lags. Time lags are important to consider since

266       information transfer is not instantaneous, but instead occurs over a span of time in a

267       biological system. By taking a sum over time lags, all edges are put on the same 'time

268       frame', and are thus more comparable.

269           Edges with the highest weights, implying greatest information transfer, include

270       (FKH1 → HOF1, SLTE=4.79), (SWI4 → HTB1, SLTE=4.75), (FKH2 → IRC8,

271       SLTE=4.47), (SWI4 → SWI1, SLTE 4.34) and (NDD1 → AIM20, SLTE=4.30). The

272       source nodes are well-known, multi-functional, cell cycle related transcription factors.

273       The target nodes have more focused functions. HOF1 regulates portions of the actin

274       cytoskeleton. HTB1 is a histone core protein required for chromatin assembly. IRC8 is a

275 bud tip localized protein, with unknown function. SWI1 is a subunit of the SWI/SNF  
276 chromatin-remodeling complex. Finally, AIM20 has unknown function, but over-  
277 expression leads to the arrest or delay of the cell cycle.

278         Some well known cell cycle regulators, such as BAS1, were not selected by any  
279 centrality measure. As far as influence is concerned, this can be explained by exploring  
280 its neighborhood in the network. In the SLTE network, BAS1 is a source to four other  
281 genes, all with no influence ranking and subgraph centrality of 0. Among the four targets  
282 is PHD1, also a target from XBP1, which does happen to be a ranked influencer, and  
283 happens to have degree 32 and a high subgraph centrality (26.8). So, although BAS1 is  
284 probably a cell cycle regulator, there are better sources to choose when targeting the  
285 same downstream genes. XBP1 binds cyclin gene promoters and is stress related.  
286 Interestingly, it's important in G1 arrest, which relates to the synchronization method  
287 used by Eser et al. in the data generation. XBP1 is also a member of the SWI4 and  
288 MBP1 protein family. While BAS1 is involved in biosynthesis pathways for histidine,  
289 purine, and pyrimidines, and predicted to be involved in mitotic crossover, XBP1, with  
290 regard to cell cycle, is certainly understandable in it's influence ranking. No one  
291 centrality metric was ideally suited towards picking out cell cycle related genes,  
292 although the ranked influencers 'pointed' to a large proportion of genes selected by  
293 other centrality measures.

294         When we considered the ranking of influential genes, we saw that high-ranking  
295 genes were more likely to be ranked high by other centrality metrics. There are several  
296 notable exceptions, where REB1, SWI4, and SWI6 were relatively low ranked  
297 influencers, but were highly ranked by other metrics. These examples are notable due

298 to their previously known role in the cell cycle and regular inclusion in models. Proteins  
299 SWI4 and SWI6 are members of the SBF complex, interacting with the MBF complex  
300 (SWI6-MBP1) to regulate late G1 events. REB1, essential in some yeast strains, is  
301 shown to act as a link between rDNA metabolism and cell cycle control in response to  
302 nutritional stress [27], and is shown to have a significant impact on lifespan [28]. The  
303 influence ranking was due to higher ranked influencers being upstream of the three  
304 genes in the regulatory network. Therefore, they were only selected as  $K$ , the set of  
305 requested influencers, grew large enough.

306 Gene set enrichment showed functions related to not only cell cycle, but also  
307 chromatin remodeling, stress response, and metabolism (see SI). The top two most  
308 influential genes, FKH1 and GCN5 have both been related to life span [29,30,31].

309 Network control is one goal in the study of dynamic networks [32,33]. Given that  
310 influential nodes seem to have a topologically advantageous position, one could  
311 speculate that influential genes might be useful selections for network control. Biological  
312 events that impact the influential nodes, thereby affecting normal information flow, could  
313 have a strong effect on the network, potentially leading to disease states. Discovering  
314 the minimum sets of biological entities that hold the greatest influence in the network  
315 context could lead to further understanding of how network dynamics is associated with  
316 disease.

## 317 ***Materials and Methods***

318 The methods described here have been implemented in python and are freely  
319 available. Run times are kept low by computing the diffusion using sparse matrix linear  
320 solvers, and using a multicore-parallel strategy for performing ant optimization. The

321 network weighting, optimization, and diffusion methods are independent, allowing  
322 researchers to "mix-and-match" their favorite modules.

### 323 **Data sources**

324 Eser et al. [23] generated time series expression data from two replicates of  
325 synchronized yeast producing metabolically labeled RNA levels every five minutes over  
326 41 time points. The expression series spans three cell cycles, which progressively  
327 dampen in wave amplitude, as yeast synchrony is lost. Using a model for detecting  
328 periodicity in gene expression, 479 genes were labeled as statistically periodic.  
329 Additionally, 32 transcription factors were predicted to be cell cycle regulators.

330 YeastMine, the database of genetic regulatory interactions in yeast (May 2015)  
331 [22] provided regulatory edges. Using 6,417 yeast genes, 33,809 genetic regulatory  
332 edges were collected. Edge weights were computed using a variation of transfer  
333 entropy, as described below.

### 334 **Computing weights using transfer entropy and time-lagged Pearson correlation**

335 Given two genes are connected by an edge, the edge weight was computed in  
336 two ways. First, time-lagged Pearson correlation was used with time lags of 0 to 6 steps  
337 (0 to 30 mins.), keeping the maximum. Second, a new variation on time-lagged transfer  
338 entropy was used similar to what is described in [34,35], termed sum-lagged transfer  
339 entropy (SLTE). TE is computed at each time lag and a sum is taken over the set of  
340 time lags. This method avoids making a choice about what time lag to use. Additionally,  
341 edge weights in the graph are composed of summed time lags, making them directly  
342 comparable. Information transfer, in the genetic regulatory context, is relatively slow and  
343 takes place over multiple time steps (each step corresponding to 5 minutes).

344 Time-lagged Pearson correlation is computed by taking two time series, or  
345 numeric vectors  $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$  and  $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ , and computing the correlation  
346 on sub-sequences  $\{x_{1+k}, \dots, x_{n-1}, x_n\}$  and  $\{y_1, y_2, \dots, y_{n-k}\}$ , where  $k$  is some integer  
347 representing the time lag between variables.

348 Transfer entropy (TE) is an information theoretic quantity that uses sequence or  
349 time series data to measure the magnitude of information transfer between variables  
350 [36,37]. Transfer entropy is model-free, directional, and shown to be related to Granger  
351 causality [38]. In TE, given two random variables  $X$  and  $Y$ , where  $X$  is directionally  
352 connected to  $Y$  (or  $X \rightarrow Y$ ), we would like to know if prior states of  $X$  help in the  
353 prediction of  $Y$ , beyond knowing the prior states of  $Y$ . With some simplifications, transfer  
354 entropy is straightforward to compute.

355 Given two sequences  $\mathbf{x}$  and  $\mathbf{y}$ , we describe transfer entropy as

$$T_{x \rightarrow y}^{(t)} = \sum_{y_t, y_{t-1}, x_{t-k}} P(y_t, y_{t-1}, x_{t-k}) \log \frac{P(y_t, y_{t-1}, x_{t-k})P(y_{t-1})}{P(y_t, x_{t-k})P(y_t, y_{t-1})},$$

356 where  $x_{t-k}$  indicates value of the sequence at time step  $t$ , with time lag  $k$ .

357 To perform the computation,  $\mathbf{x}$  and  $\mathbf{y}$  are normalized and used to fit a Gaussian  
358 kernel density estimate (KDE) with the default adaptive bandwidth. Then, by sampling  
359 the density estimate, and normalizing the samples, a three-dimensional grid  
360 representing the joint probability is generated. The required distributions are  
361 marginalized from the joint distribution. Smaller grid sizes provide a finer grained  
362 probability distribution, but greatly slow the computation without changing the values  
363 substantially. A three dimensional grid of  $10^3$  points was found to be a good  
364 compromise between computation time and accuracy.

365 A permutation test is performed to assess statistical significance of the transfer  
366 entropy,  $T_{x \rightarrow y}$ . The sequence  $x$  is permuted over some number of trials and a count is  
367 taken on the number of times the permuted TE is greater than the observed TE, giving  
368 an empirical p-value.

### 369 **The diffusion model is used to score solutions to the IMP**

370 The information flow model, and most of the nomenclature, is described in [12].  
371 The diffusion models are Markov chains with absorbing states [39]. In the model,  
372 vertices are first partitioned into sets S and T. The set S contains sources, generating  
373 information, which then flow through the network (nodes in T) until reaching a dead end  
374 or absorbing back into S.

375 The stochastic matrix -- defining the probability of moving from one vertex to  
376 another-- is defined as

$$p_{ij} = \frac{w_{ij}}{\sum_j w_{ij}},$$

377 where edge weights  $w_{ij}$  are the weights on outgoing edges. Sets S and T partition the  
378 stochastic matrix as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{SS} & \mathbf{P}_{ST} \\ \mathbf{P}_{TS} & \mathbf{P}_{TT} \end{bmatrix}$$

379 where  $\mathbf{P}_{SS}$  defines the transition probabilities from nodes in S to S, and  $\mathbf{P}_{ST}$   
380 defines transition probabilities from S to T, and so on. Although the matrix is square, it is  
381 not symmetric, given the directed edges.

382 Ultimately, we wish to compute the expected number of visits from a node  $v_i \in S$ ,  
383 to a node  $v_j \in T$ , defined as matrix  $\mathbf{H}$ . By time point  $t$ , the estimated number of visits  
384 from  $v_i \in S$  to  $v_j \in T$  is found as

$$h_{ij}^{(t)} = p_{ij} + \sum_{k \in T} h_{jk}^{(t-1)} p_{ki}$$

385 where  $p_{ij}$  is the transition probability of  $v_i \in S$  to  $v_j \in T$  and  $h_{jk}^{(t-1)}$  is the expected  
386 number of visits that have already taken place at time  $(t - 1)$ , from  $v_i \in S$  to  $v_j \in T$ . At  
387 time step  $t$ , information can travel from  $v_i \in S$  to  $v_j \in T$  directly, or it would already be at  
388 adjacent node  $v_k$ , and would travel from  $v_k \in T$  to  $v_j \in T$  in the next time step. The  
389 matrix form of the equation is given as

$$\mathbf{H}^{(t)} = \mathbf{P}_{ST} + \mathbf{H}^{(t-1)}\mathbf{P}_{TT}.$$

390 In the long run, at steady state, when  $\mathbf{H}^{(t)} \sim \mathbf{H}^{(t-1)}$ , the equation reduces to  
391  $\mathbf{H}(\mathbf{I} - \mathbf{P}_{TT}) = \mathbf{P}_{ST}$ , where  $\mathbf{I}$  is the identity matrix. By taking the transpose of both sides,  
392 we have  $(\mathbf{I} - \mathbf{P}_{TT})'\mathbf{H}' = \mathbf{P}'_{ST}$ . This form lets us avoid the matrix inverse in solving for  $\mathbf{H}$ ,  
393 which can be expensive to compute, and lets us use iterative solvers that can even  
394 handle singular matrices. Specifically, the Python SciPy sparse linear algebra library  
395 has solvers appropriate for this problem.

396 To compute a measure of influence on the network, after solving for  $\mathbf{H}$  the  
397 expected number of visits on nodes, the influence is summarized as the “influence-  
398 score”,

$$InfluenceScore = \sum_{i \in S} \left\{ \sum_{j \in T} I(h_{ij} > \theta) \right\} + w_i$$

399 where  $h_{ij}$  is the number of visitations (using matrix  $\mathbf{H}$ ) from node  $v_i \in S$  to connected  
400 nodes  $v_j \in T$ . Indicator function  $I(h_{ij} > \theta)$  is equal to 1 if the number visitations is  
401 greater than a threshold  $\theta$  and  $w_i$  is the sum of outgoing weights for node  $v_i \in S$ . The  
402 sum of edge weights is used as a tie-breaker in the case of degenerate solutions, and

403 also makes the case for a solution set that is best supported by data. This influence  
404 score is the equivalent to computing the cover on nodes in T. In this work,  $\theta = 0.0001$  is  
405 used.

#### 406 **Ant optimization is used to search for influential nodes**

407 An implementation of the hypercube min-max ant optimization algorithm was  
408 constructed to search for solutions to the Influence Maximization Problem [40,41]. Ant  
409 optimization is based on the idea of probabilistically constructing potential solutions to a  
410 given problem, in this case a subset selection problem, and reinforcing good solutions  
411 with a "pheromone" weight deposited on solution components, ensuring that good  
412 solutions become increasingly likely in later iterations.

413 Since the algorithm is stochastic and results can vary, the optimization is  
414 repeated for a defined number of runs. Each convergence takes a number of iterations  
415 where ants construct solutions, perform a local search, score the solutions using the  
416 influence score, and reinforce the components. As a run progresses, the pheromone  
417 values move to either one or zero, indicating whether the component was selected.

418 At the start of each iteration, ants construct potential solutions, a subset of  
419 vertices, by sampling from nodes using probability distribution

$$q_i = \frac{u_i^\alpha r_i^\beta}{\sum u_i^\alpha r_i^\beta}$$

420 where  $q_i$  is the probability for sampling any node  $v_i$ , with the sum of outgoing  
421 edges giving node weight  $u_i$  and pheromone weight  $r_i$ . The alpha and beta parameters  
422 are used to give importance to either node weights or pheromones. Solutions are  
423 constructed by sampling one node at a time. After each sample, the probabilities are

424 renormalized. Here,  $\alpha$  and  $\beta$  are set to 1.

425         Local search is performed by stochastic hill climbing, where random, alternative  
426 solutions are tried. If a better score is found, the solution is replaced, and carried  
427 forward. Local search has a strong effect on the quality of the solutions, and even a  
428 small number of hill climbing steps tends reduce the time required for convergence.

429         Next, using the influence score function, each potential solution is scored, with  
430 the best solution kept and compared to solutions found in earlier runs. As part of the  
431 Min-Max algorithm, three solutions are kept throughout the run: the iteration-best, the  
432 restart-best and the overall-best. The pheromone updates use a weighted average over  
433 the three solutions. At the beginning of the run, the pheromone updates are entirely  
434 from the iteration-best solution, but gradually, the updates are increasingly influenced by  
435 the restart and overall-best solutions, which is done to avoid local minima. The weighted  
436 average pheromone would be  $r_{avg} = f_1 b_i + f_2 b_r + f_3 b_b$  where  $b_i$  is the iteration best,  $b_r$   
437 is the restart best,  $b_b$  is the best overall, and fractions  $f_1 + f_2 + f_3 = 1$ . The pheromone  
438 updates are defined as  $r^{(t+1)} = r^{(t)} + d(r_{avg} - r^{(t)})$ , where  $r^{(t)}$  is the pheromone  
439 weights at time  $t$ ,  $d$  is the learning rate, and  $r_{avg}$  is the average over the three solutions.  
440 Eventually, the pheromone weights become sufficiently close to zero or one, and if all  
441 runs are complete, the solution is returned with the influence score.

#### 442 **Additional 'off-the-shelf' analysis**

443         BioFabric, R and R packages igraph, pheatmap and ggplot2 were used for  
444 visualization and analysis [42,43,44,49]. Cytoscape 3.3.0 was used for visualizing  
445 graphs [45,46]. Pathway and GO term enrichment was generated using the CPDB from  
446 The Max Planck Institute for Molecular Genetics [47]. SciPy was used in the software

447 implementation [48].

## 448 **Acknowledgements**

449 Thanks to the Shmulevich Lab and the Institute for Systems Biology.

## 450 **References**

- 451 1. Waltermann C, Klipp E. Information theory based approaches to cellular signaling. *Biochim Biophys*  
452 *Acta*. 2011;1810(10):924–32.
- 453
- 454 2. Nurse P. Life, logic and information. *Nature*. 2008; 454:424-426
- 455
- 456 3. Missiuro PV, Liu K, Zou L, Ross BC, Zhao G, Liu JS, et al. Information flow analysis of interactome  
457 networks. *PLoS Comput Biol*. 2009 Apr;5(4):e1000350.
- 458
- 459 4. Vandin F, Clay P, Upfal E, Raphael BJ. Discovery of mutated subnetworks associated with clinical data  
460 in cancer. *Pac Symp Biocomput*. 2012;55–66.
- 461
- 462 5. Basha O, Tirman S, Eluk A, Yeger-Lotem E. ResponseNet2.0: Revealing signaling and regulatory  
463 pathways connecting your proteins and genes--now with human data. *Nucleic Acids Res*. 2013;41:198–  
464 203.
- 465
- 466 6. Morone F, Makse H. Influence maximization in complex networks through optimal percolation. *Curr Sci*.  
467 2015;93(1):17–9.
- 468
- 469 7. Singer Y. How to Win Friends and Influence People, Truthfully: Influence Maximization Mechanisms for  
470 Social Networks. *Fifth ACM Int Conf Web Search Data Min*. 2012;1–10.
- 471
- 472 8. Domingos P, Richardson M. Mining the Network Value of Customers. *Proc Seventh ACM SIGKDD Int*  
473 *Conf Knowl Discov Data Min*. 2001;57–66
- 474
- 475 9. Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. *Proc*  
476 *ninth ACM SIGKDD Int Conf Knowl Discov data Min - KDD '03*. 2003;137.
- 477
- 478 10. Kitsak M, Gallos LK, Havlin S, Liljeros F, Muchnik L, Stanley HE, et al. Identifying influential spreaders  
479 in complex networks. *Nat Phys*. 2010;6(11):36.
- 480
- 481 11. Dawson DA. Information flow in graphs. *Stoch Process their Appl*. Elsevier; 1975;3(2):137–51.
- 482
- 483 12. Stojmirović A, Yu YK. Information flow in interaction networks. *J Comput Biol*. 2007;14(8):1115–43.
- 484
- 485 13. Kim Y-A, Przytycki JH, Wuchty S, Przytycka TM. Modeling information flow in biological networks.  
486 *Phys Biol*. IOP Publishing; 2011;8(3):035012.
- 487
- 488 14. Leguizamon G, Michalewicz Z. A new version of ant system for subset problems. *Proc 1999 Congr*.  
489 1999;
- 490
- 491 15. Solnon C, Bridge D. An ant colony optimization meta-heuristic for subset selection problems. *Systems*  
492 *Engineering Using Particle Swarm Optimisation*. 2007
- 493
- 494 16. Verwaeren J, Scheerlinck K, De Baets B. Countering the negative search bias of ant colony  
495 optimization in subset selection problems. *Comput & Oper*. 2013;

- 496  
497 17. Yang WS, Weng SX. Application of the Ant Colony Optimization Algorithm to the Influence-  
498 Maximization Problem. *Int J Swarm Intell Evol Comput.* 2012;1:1–8.  
499
- 500 18. Zotenko E, Mestre J, O'Leary DP, Przytycka TM. Why Do Hubs in the Yeast Protein Interaction  
501 Network Tend To Be Essential: Reexamining the Connection between the Network Topology and  
502 Essentiality. *PLoS Comput Biol. Public Library of Science*; 2008 Aug;4(8):e1000140.  
503
- 504 19. Newman MEJ. A measure of betweenness centrality based on random walks. *Social Networks.* 2003  
505 Sep;27(1):39-54.  
506
- 507 20. Page L, Brin S, Motwani R, Winograd T. The PageRank Citation Ranking: Bringing Order to the Web.  
508 - Stanford InfoLab Publication Server. 1999;  
509
- 510 21. Haase SB, Wittenberg C. Topology and control of the cell-cycle-regulated transcriptional circuitry.  
511 *Genetics.* 2014 Jan;196(1):65–90.  
512
- 513 22. Balakrishnan R, Park J, Karra K, Hitz BC, Binkley G, Hong EL, et al. YeastMine--an integrated data  
514 warehouse for *Saccharomyces cerevisiae* data as a multipurpose tool-kit. *Database (Oxford).* 2012 Jan;  
515 bar062.  
516
- 517 23. Eser P, Demel C, Maier KC, Schwalb B, Pirkel N, Martin DE, et al. Periodic mRNA synthesis and  
518 degradation co-operate during cell cycle gene expression. *Mol Syst Biol.* 2014 Jan;10(1):717.  
519
- 520 24. Ramsey SA, Klemm SL, Zak DE, Kennedy KA, Thorsson V, et al. Uncovering a Macrophage  
521 Transcriptional Program by Integrating Evidence from Motif Scanning and Expression Dynamics. *PLoS*  
522 *Comput Biol* 2008;4(3): e1000021.  
523
- 524 25. Clauset A, Shalizi CR, Newman MEJ. Power-Law Distributions in Empirical Data. *SIAM Rev.*, 51(4),  
525 661–703.  
526
- 527 26. Kraikivski, P, Chen, KC, Laomettachit, T, Murali, TM, and Tyson, JJ. From START to FINISH:  
528 computational analysis of cell cycle control in budding yeast. *NPJ Syst. Biol. Appl.* 2015;1:15016.  
529
- 530 27. Leonor Rodríguez-Sánchez, María Rodríguez-López, Zaira García, María Tenorio-Gómez, Jorge B.  
531 Schvartzman, Dora B. Krimer, Pablo Hernández. The fission yeast rDNA-binding protein Reb1 regulates  
532 G1 phase under nutritional stress. *J Cell Sci.* 2011;124: 25-34  
533
- 534 28. Kamei Y, et al. Transcription factor genes essential for cell proliferation and replicative lifespan in  
535 budding yeast, *Biochem. Biophys. Res. Commun.* 2015; 463(3):351-356.  
536
- 537 29. McCormick MA, et al. The SAGA histone deubiquitinase module controls yeast replicative lifespan via  
538 Sir2 interaction. *Cell Rep.* 2014 Jul 24;8(2):477-86.  
539
- 540 30. Grant PA, et al. Yeast Gcn5 functions in two multisubunit complexes to acetylate nucleosomal  
541 histones: characterization of an Ada complex and the SAGA (Spt/Ada) complex. *Genes Dev.* 1997 Jul  
542 1;11(13):1640-50.  
543
- 544 31. Postnikoff SD, Malo ME, Wong B, Harkness TA. The yeast forkhead transcription factors fkh1 and  
545 fkh2 regulate lifespan and stress response together with the anaphase-promoting complex. *PLoS Genet.*  
546 2012;8(3):e1002583.  
547
- 548 32. Cowan NJ, Chastain EJ, Vilhena DA, Freudenberg JS, Bergstrom CT. Nodal Dynamics, Not Degree  
549 Distributions, Determine the Structural Controllability of Complex Networks. *PLoS One. Public Library of*  
550 *Science*; 2012 Jun;7(6):e38398.  
551

- 552 33. Onnela JPJ. Flow of Control in Networks. *Sci*. 2014 Mar;343(6177):1325–6.  
553
- 554 34. Wibral M, Pampu N, Priesemann V, Siebenhühner F, Seiwert H, Lindner M, et al. Measuring  
555 Information-Transfer Delays. *PLoS ONE*. 2013;8(2): e55809.  
556
- 557 35. Faes L, Marinazzo D, Montalto A, Nollo G. Lag-specific transfer entropy as a tool to assess  
558 cardiovascular and cardiorespiratory information transfer. *IEEE Trans Biomed Eng*. 2014  
559 Oct;61(10):2556–68.  
560
- 561 36. Schreiber T. Measuring information transfer. *Phys Rev Lett*. 2000 Jul;85(2):461–4.  
562
- 563 37. Lee J, Nemati S, Silva I, Edwards BA, Butler JP, Malhotra A. Transfer Entropy Estimation and  
564 Directional Coupling Change Detection in Biomedical Time Series. *Biomed Eng Online*. 2012 Apr  
565 13;11:19.  
566
- 567 38. Hlaváčková-Schindler K. Equivalence of granger causality and transfer entropy: A generalization.  
568 *Appl Math Sci*. 2011;5(73):3637-3648.  
569
- 570 39. Kemeny JG, Snell JL. *Finite markov chains*. Princeton, NJ: van Nostrand; 1960.  
571
- 572 40. Stutzle T, Hoos HH. MAX-MIN ant system. *Futur Gener Comput Syst*. Elsevier; 2000;16(8):889–914.  
573
- 574 41. Blum C, Dorigo M. The Hyper-Cube Framework for Ant Colony Optimization. *IEEE Trans Syst Man  
575 Cybern B Cybern*. 2004 Apr;34(2):1161–72.  
576
- 577 42. Csardi G, Nepusz T. The igraph Software Package for Complex Network Research. *InterJournal*.  
578 2006;Complex Sy:1695.  
579
- 580 43. Kolde R. *heatmap: Pretty Heatmaps*. R package version 1.0.7. 2015  
581
- 582 44. Wickham H. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. 2009.  
583
- 584 45. Shannon P. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction  
585 Networks. *Genome Res [Internet]*. 2003 Nov;13(11):2498–504.  
586
- 587 46. Smoot ME, Ono K, Ruscheinski J, Wang PL, Ideker T. Cytoscape 2.8: new features for data  
588 integration and network visualization. *Bioinformatics*. 2011 Jan;27(3):431–2.  
589
- 590 47. Kamburov A, Pentchev K, Galicka H, Wierling C, Lehrach H, Herwig R. ConsensusPathDB: toward a  
591 more complete picture of cell biology. *Nucleic Acids Res*. 2011 Jan;39(Database issue):D712-7.  
592
- 593 48. Stéfan W, Colbert SC, Varoquaux G. The NumPy Array: A Structure for Efficient Numerical  
594 Computation, *Comput Sci Eng*. 2011;13:22-30.  
595
- 596 49. Longabaugh, WJR. Combing the hairball with BioFabric: a new approach for visualization of large  
597 networks. *BMC Bioinformatics*. 2012;13:275.

Figure 1

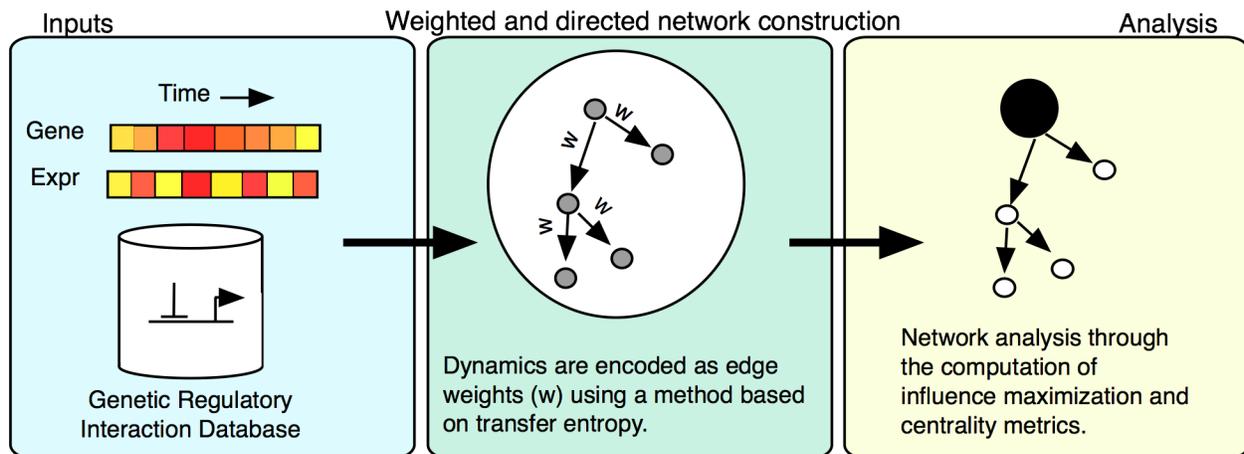


Figure 2

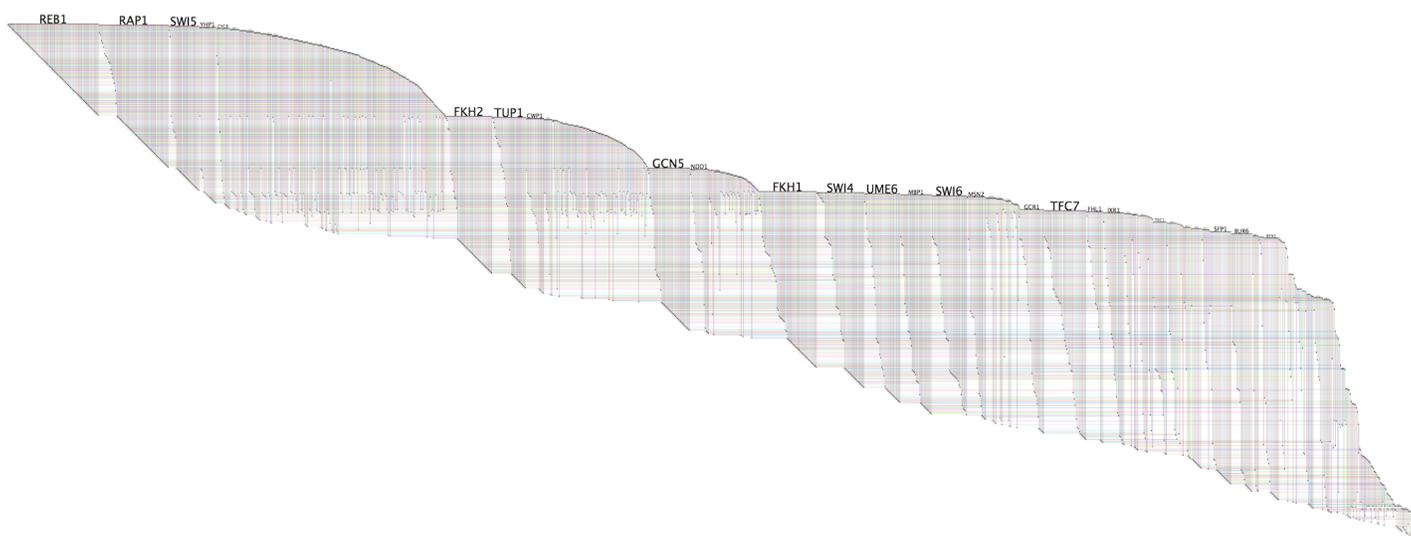


Figure 3

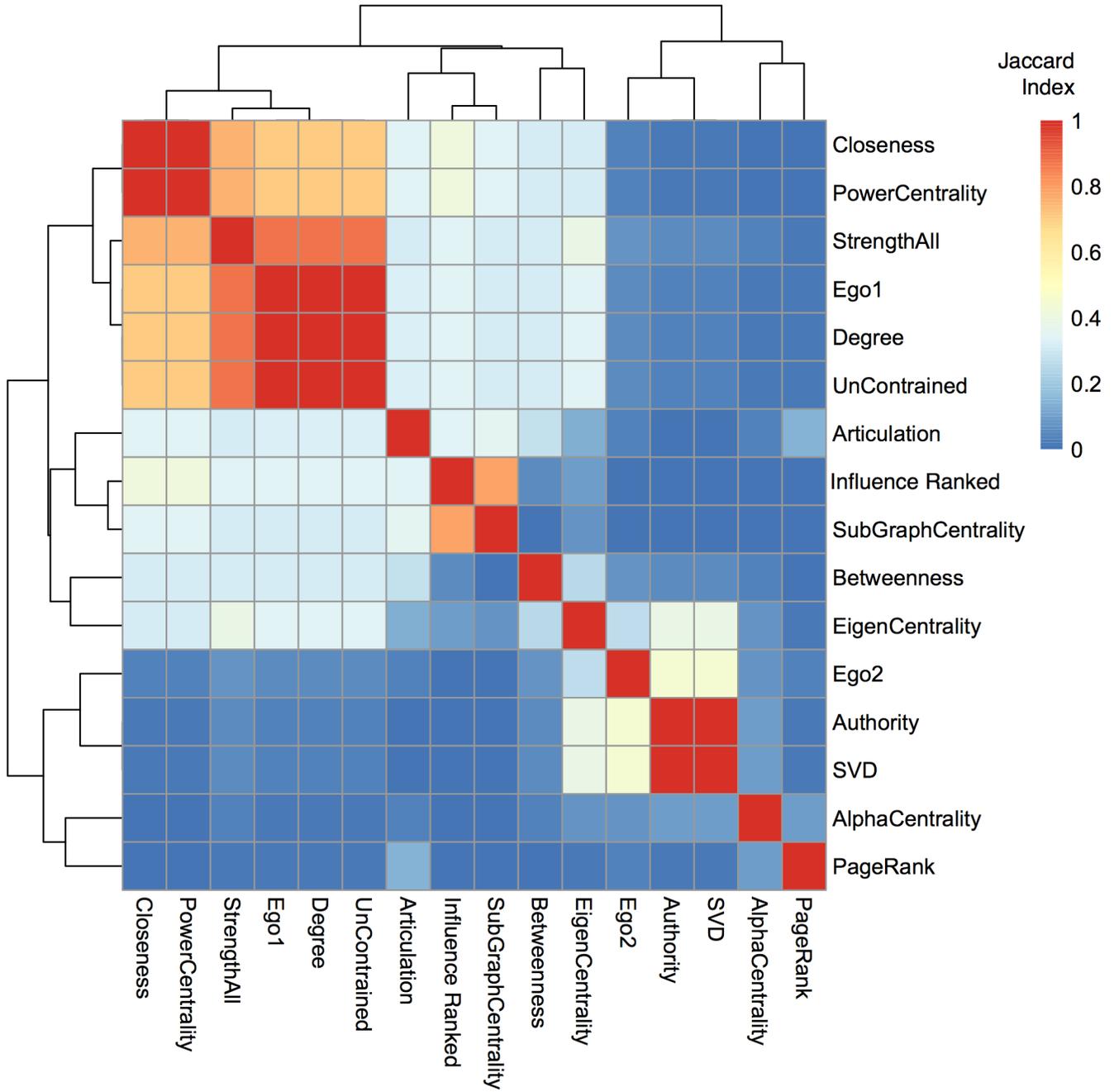


Figure 4

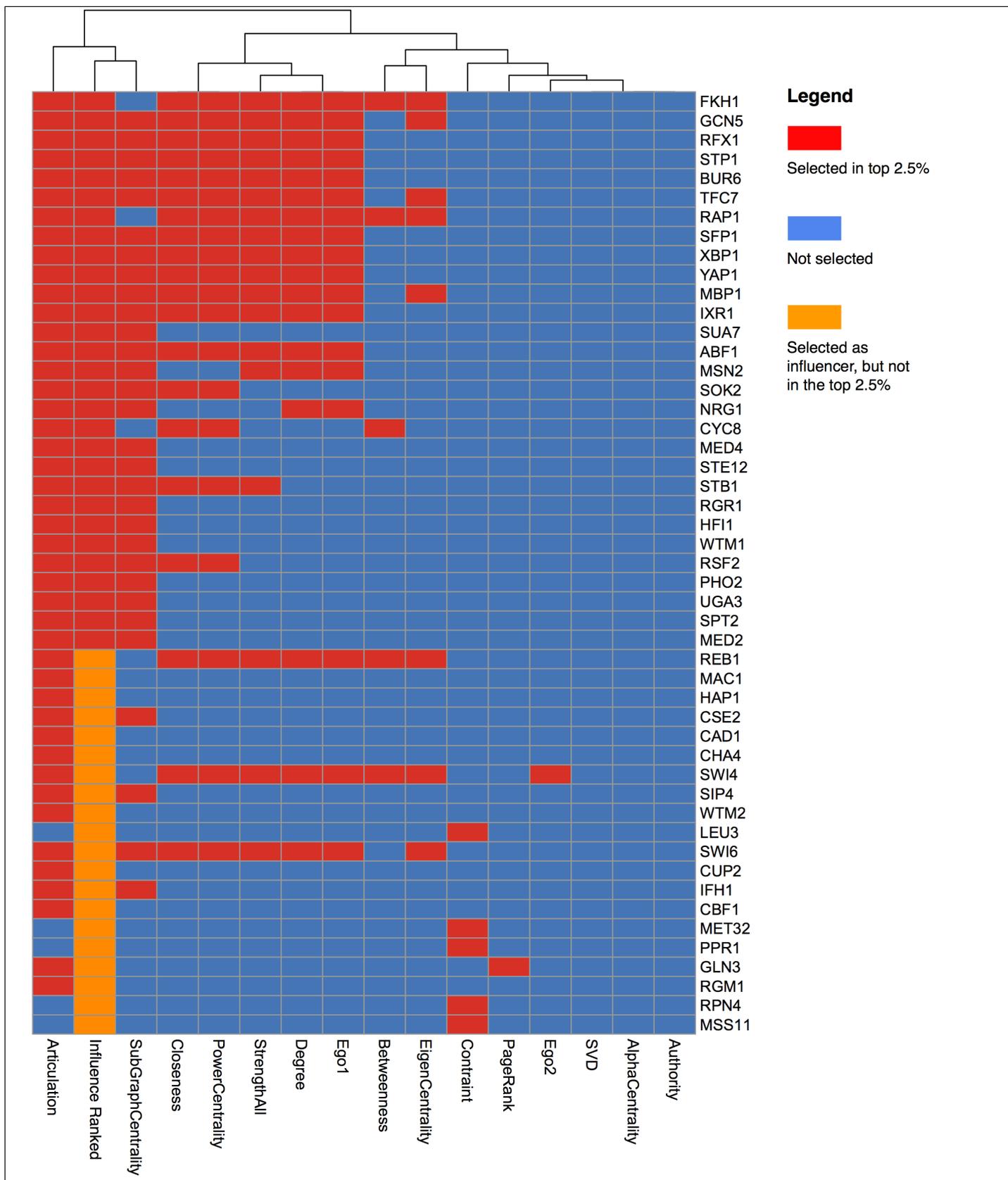




Figure 6

