Inferring network mechanisms: The *Drosophila melanogaster* protein interaction network

Manuel Middendorf[†], Etay Ziv[‡], and Chris H. Wiggins[§¶‖]

[†]Department of Physics, [‡]College of Physicians and Surgeons, [§]Department of Applied Physics and Applied Mathematics, and [¶]Center for Computational Biology and Bioinformatics, Columbia University, New York, NY 10027

Communicated by Barry H. Honig, Columbia University, New York, NY, December 20, 2004 (received for review September 7, 2004)

Naturally occurring networks exhibit quantitative features revealing underlying growth mechanisms. Numerous network mechanisms have recently been proposed to reproduce specific properties such as degree distributions or clustering coefficients. We present a method for inferring the mechanism most accurately capturing a given network topology, exploiting discriminative tools from machine learning. The *Drosophila melanogaster* protein network is confidently and robustly (to noise and training data subsampling) classified as a duplication–mutation–complementation network over preferential attachment, small-world, and a duplication–mutation mechanism without complementation. Systematic classification, rather than statistical study of specific properties, provides a discriminative approach to understand the design of complex networks.

machine learning | systems biology | motifs | classification | evolution

Recent research activity in biological networks has often focused on understanding the emergence of specific features such as scale-free degree distributions (1–3), short mean geodesic lengths, or clustering coefficients (4). The insights gained into the topological patterns have motivated various network growth and evolution models to determine what simple mechanisms can reproduce the features observed. Among these are the preferential attachment model (3, 5), exhibiting scale-free degree distributions, and the small-world model (4), exhibiting high clustering coefficients despite short mean geodesics. Additionally, various duplication–mutation mechanisms have been proposed to describe biological networks (6–11) and the World Wide Web (12). However, in most cases model parameters can be tuned such that multiple models of widely varying mechanisms perfectly fit the motivating real network in terms of single selected features such as the scale-free exponent and the clustering coefficient (compare Fig. 1). Because networks with several thousands of vertices and edges are highly complex, it is also clear that these statistics can capture only limited structural information.

Here, we make use of *discriminative classification* techniques recently developed in machine learning (13, 14) to classify a given real network as one of many proposed network mechanisms by enumerating local substructures. Determining what simple mechanism is responsible for a natural network's architecture (*i*) facilitates the development of correct priors for constraining network inference and reverse engineering (15–18); (*ii*) specifies the appropriate null model relative to which one evaluates statistical significance (19–29); (*iii*) guides the development of improved network models; and (*iv*) reveals underlying design principles of evolved biological networks. It is therefore desirable to develop a method to determine which proposed mechanism models a given complex network without prior selection of features or null models.

Enumeration of subgraphs has been successfully used in the past few years to find network motifs (19, 20, 23–29) and is historically a well established method in the sociology community (30–32). Recently, the idea of clustering real networks based on their "significance profiles" has been proposed (33). The method assesses significance of given subgraphs relative to an assumed null model, generated by Monte Carlo sampling of networks with a degree distribution identical to that of the network of interest. The significance profiles are then shown to be similar for various groups of naturally occurring networks.

Both clustering and assessing statistically significant motifs can be characterized as schemes to identify reduced-complexity descriptions of the networks. We here present an approach that is instead *predictive,* using labeled graphs of known growth mechanisms as training data for a discriminative classifier. This classifier, then, presented with a new graph of interest, can reliably and robustly predict the growth mechanism that gave rise to that graph. Within the machine learning community, such predictive, *supervised learning*, techniques are differentiated from descriptive, *unsupervised learning*, techniques such as clustering.

We apply our method to the recently published *Drosophila melanogaster* protein–protein interaction network (34) and find that a duplication–mutation–complementation (DMC) mechanism (6) best reproduces *Drosophila*'s network. The prediction is robust against noise, even after random rewiring of up to 45% of the network edges. To validate, we also show that beyond 80% random rewiring the correct (Erdös–Rényi) classification is obtained.

Methods

The Data Set. We use a protein–protein interaction map based on yeast two-hybrid screening (34). Because the data are subject to numerous false positives, Giot *et al.* (34) assign a confidence score $P \in [0, 1]$, measuring how likely the interaction occurs *in vivo*. To exclude unlikely interactions and focus on a core network that retains significant global features, we determine a confidence threshold p^* based on percolation: measurements of the size of the components for all possible values of p^* show that the two largest components are connected for $p^* = 0.65$ (see the supporting information, which is published on the PNAS web site). Edges in the graph correspond to interactions for which $p > p^*$. To reveal possible structural changes in *Drosophila* for less stringent thresholds, we also present results for $p^* = 0.5$ as suggested in ref. 34. We remove self-interactions from the network because none of the proposed mechanisms allow for them. After eliminating isolated vertices the resulting networks consist of 3,359 (4,625) vertices and 2,795 (4,683) edges for $p^* = 0.65$ (0.5).

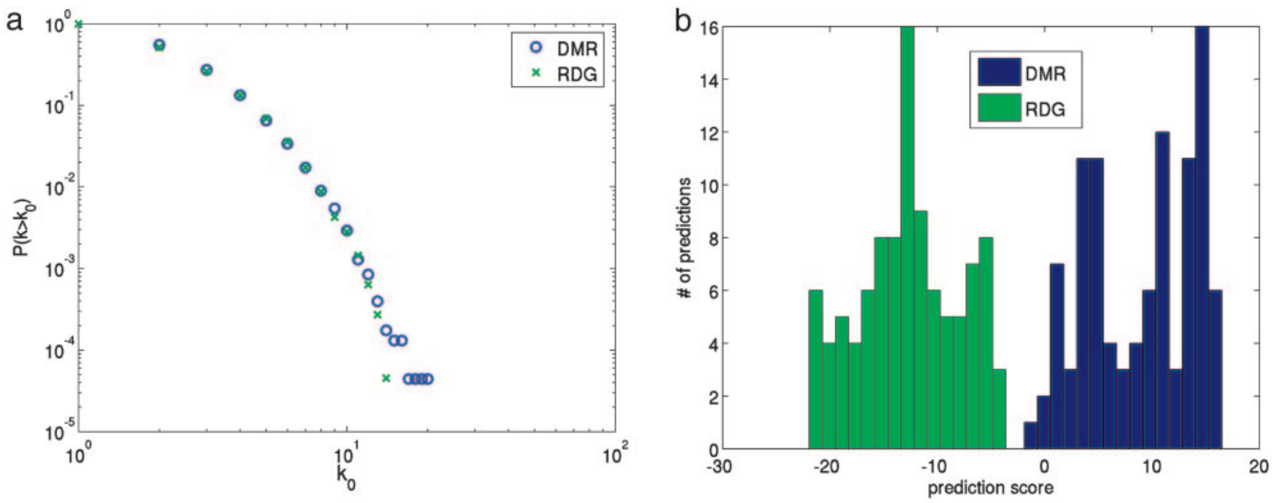
Network Mechanisms. We generate 7,000 graphs, 1,000 for each of seven different models drawn from the literature, as training

Abbreviations: DMC, duplication–mutation–complementation; DMR, duplication–mutation using random mutations; LPA, linear preferential attachment; RDS, random static; RDG, random growing; AGV, aging vertex; SMW, small-world; ADT, alternating decision tree.

See Commentary on page 3173.

[‖]To whom correspondence should be addressed at: Department of Applied Physics and Applied Mathematics, Columbia University, 500 West 120th Street, New York, NY 10027. E-mail: chris.wiggins@columbia.edu.

www.pnas.org/cgi/doi/10.1073/pnas.0409515102



	DMR	RDG
$\langle C \rangle$	$2.6\ 10^{-4} \pm 1.3\ 10^{-4}$	$5.4\ 10^{-4} \pm 3.7\ 10^{-4}$
$\langle \ell \rangle$	10.4 ± 0.1	9.6 ± 0.04

Fig. 1. Discriminating similar networks. Ten graphs of two different mechanisms exhibit similar average geodesic lengths and almost identical degree distribution and clustering coefficients. (*a*) Cumulative degree distribution $p(k > k_0)$, average clustering coefficient $\langle C \rangle$ and average geodesic length $\langle \ell \rangle$, all quantities averaged over a set of 10 graphs. (*b*) Prediction scores for all 10 graphs and all five cross-validated (13) ADTs. The two sets of graphs can be perfectly separated by our classifier, even though none of these graphs is used in the classifier training.

data. Every graph is generated with the same number of edges and number of vertices as measured in *Drosophila*; all other existing parameters are sampled uniformly (see supporting information). The models, many of which were explicitly intended to model protein interaction networks, manifest various simple network growth mechanisms. As an example, the DMC algorithm (6) is inspired by an evolutionary model of the genome (35, 36) proposing that most of the duplicate genes observed today have been preserved by functional complementation. If either copy of the gene loses one of its functions (edges), the other becomes essential in ensuring the organism's survival. There is thus an increased preservation of duplicate genes induced by null mutations. The algorithm features a duplication step followed by mutations that preserve functional complementarity. At every iteration one chooses a vertex v at random. A twin vertex v_{twin} is then introduced, copying all of v's edges. For each edge of v, one deletes with probability q_{del} either the original edge or its corresponding edge of v_{twin}. The twins themselves are conjoined with an independent probability q_{con}, representing an interaction of a protein with its own copy. Note that no new edges are created by mutations. The DMC mechanism thus assumes that the probability of creating new advantageous functions by random mutations is negligible.

A slightly different implementation of duplication–mutation is realized in ref. 7 by using random mutations (DMR). Possible interactions between twins are neglected. Instead, edges between v_{twin} and the neighbors of v can be removed with a probability q_{del} and new edges can be created at random between v_{twin} and any other vertices with a probability q_{new}/N, where N is the current total number of vertices. DMR thus emphasizes the creation of new advantageous functions by mutation.

In addition to (*i*) DMC and (*ii*) DMR, we generate training data for (*iii*) linear preferential attachment (LPA) networks (3, 5) (growing graphs with a probability of attaching new vertices to existing vertices proportional to $k + a$, a being a constant parameter and k being the degree of the existing vertex); (*iv*) random static (RDS) networks (37) (also known as Erdös–Rényi graphs; vertices are connected randomly); (*v*) random growing (RDG) networks (38) (growing graphs where new edges are created randomly between existing vertices); (*vi*) aging vertex (AGV) networks (39) (growing graphs modeling citation networks, where the probability for new edges decreases with the age of the vertex); and (*vii*) small-world (SMW) networks (4) (an interpolation between regular ring lattices and randomly connected graphs). For descriptions of the specific algorithms we refer the reader to the supporting information.

Subgraph Census. We quantify the topology of a network by exhaustive subgraph census (31) up to a given subgraph size; note that we do *not* assume a specific network randomization or test for statistical significance as in refs. 19, 20, 23–29, 31, and 32, but we instead *classify* network mechanisms by using the raw subgraph counts. Rather than choosing most important features *a priori*, we count all possible subgraphs up to a given cut-off, which can be made in the number of vertices, number of edges, or the length of a given walk. To show robustness to this choice, we present results for two different cut-offs. We first count all subgraphs that can be constructed by a walk of length eight (148 nonisomorphic[††] subgraphs); second, we consider all subgraphs up to a total number of seven edges (130 nonisomorphic subgraphs). Their counts are the input features for our classifier. It is worth noting that the mean geodesic length (average shortest path between two vertices) of the *Drosophila* network's giant component is 11.6 (9.4) for $p^* = 0.65$ (0.5). Walks of length eight are therefore able to traverse large parts of the network and can also reveal global structures.

Learning Algorithm. Our classifier is a generalized decision tree called an *alternating decision tree* (ADT) (40) by using the Adaboost (41) algorithm, which is related to additive logistic regression (42). Adaboost is a general discriminative learning algorithm proposed in 1997 by Freund and Schapire (41, 43) and has since been successfully used in numerous and varied applications [e.g., in text categorization (44, 45) and gene expression prediction (46)].

[††] Two graphs are isomorphic if there exists a relabeling of their vertices such that the two graphs are identical.

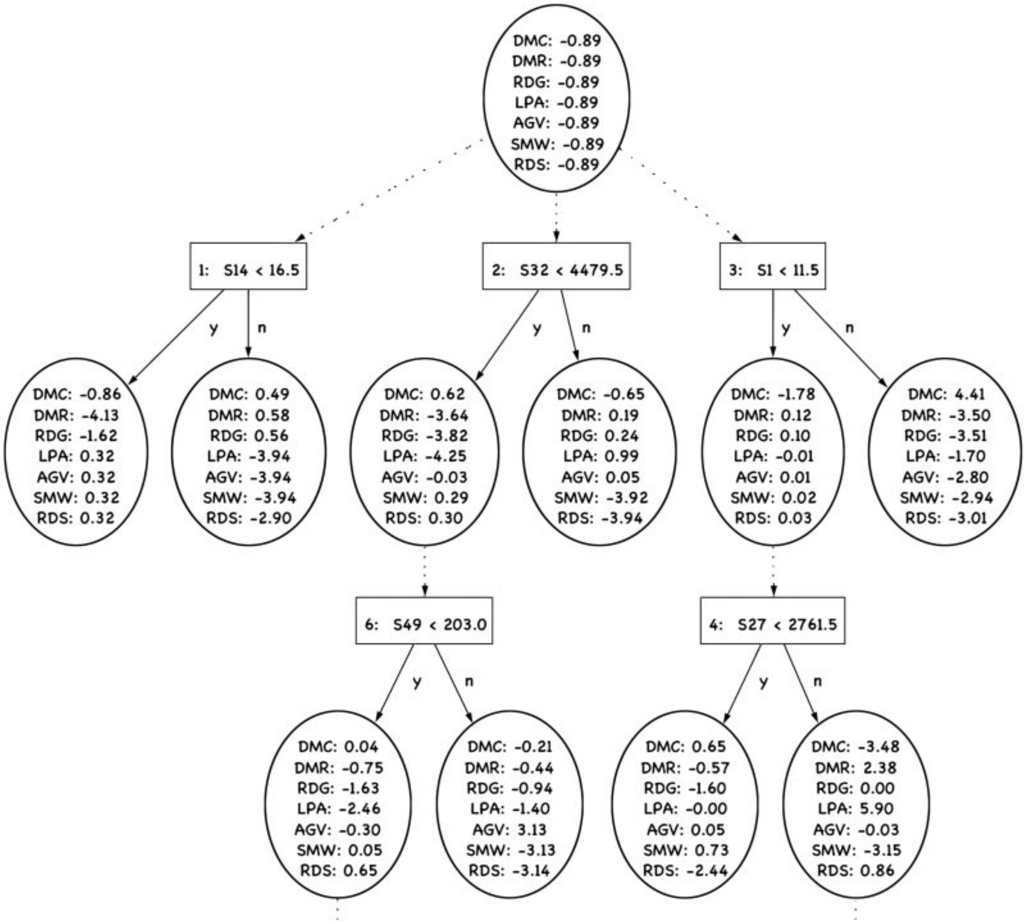


Fig. 2. ADT: The first few nodes of one of the trained ADTs are shown. At each boosting iteration one new decision node (rectangle) with its two prediction nodes (ovals) is introduced. Every test network follows multiple paths in the tree, dictated by the inequalities in the decision nodes (S# refers to a specific subgraph count; see Fig. 3). The final score is the sum of all prediction scores over all paths, and the class with the highest prediction score wins.

An example of an ADT is shown in Fig. 2. A given network's subgraph counts determine paths in the ADT dictated by inequalities specified by the *decision nodes* (rectangles) (subgraphs associated with Fig. 2 are shown in Fig. 3). For each class, the ADT outputs a real-valued *prediction score*, which is the sum of all weights over all paths. The class with the highest score wins. The prediction score $y(c)$ for class c is related to the probability $p(c)$ for the tested network to be in class c by $p(c) = e^{2y(c)}/(1 + e^{2y(c)})$ (42). (The supporting information gives additional details on the exact learning algorithm. Source code is available from C.H.W. on request.)

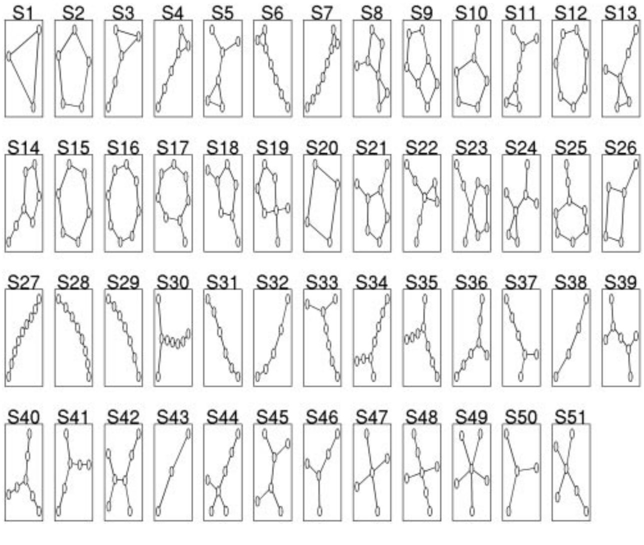


Fig. 3. Subgraphs associated with Figs. 2 and 4. Shown is the subset of 51 subgraphs (of 148) that appear in the learned ADT.

An advantage of ADTs is that they do not assume a specific geometry of the input space; that is, features are not coordinates in a metric space (as in support vector machines or k-nearest-neighbors classifiers), and the classification is thus independent of normalization. The algorithm assumes neither independence nor dependence among subgraph counts. The subgraphs reveal their importance themselves solely by their abilities to discriminate among different classes.

Results

We perform cross-validation (ref. 13 and supporting information) with multiclass ADTs, thus determining an empirical estimate of the generalization error, i.e., the probability of mislabeling an unseen test datum. Table 1 relates truth and prediction for the test sets. Five of seven classes have nearly perfect prediction accuracy. Because AGV is constructed to be an interpolation between LPA and a ring lattice, the AGV, LPA, and SMW mechanisms are equivalent in specific parameter regimes and correspondingly show a nonnegligible overlap. Nevertheless, the overall prediction accuracy on the test sets still lies between 94.6% and 95.8% for different choices of p^* and

Table 1. Prediction accuracy (%) for tested networks using fivefold cross-validation (13)

	Prediction						
Truth	DMR	DMC	AGV	LPA	SMW	RDS	RDG
DMR	99.3	0.0	0.0	0.0	0.0	0.1	0.6
DMC	0.0	99.7	0.0	0.0	0.3	0.0	0.0
AGV	0.0	0.1	84.7	13.5	1.2	0.5	0.0
LPA	0.0	0.0	10.3	89.6	0.0	0.0	0.1
SMW	0.0	0.0	0.6	0.0	99.0	0.4	0.0
RDS	0.0	0.0	0.2	0.0	0.8	99.0	0.0
RDG	0.9	0.0	0.0	0.1	0.0	0.0	99.0

The (*i, j*) entry is the probability of predicting class *j* given that the true class is *i*. The training data are based on the size of the *Drosophila* protein network with a confidence threshold of $p^* = 0.5$, the input features of the classifier being counts of all possible walks of length eight. The overall prediction accuracy is 95.8%. Prediction errors among AGV, LPA, and SMW networks are due to equivalence of the models in specific parameter regimes.

subgraph size cut-off. Note that preferential attachment is completely distinguishable from duplication–mutation despite the fact that a duplication mechanism is sometimes described as an *effective* preferential attachment (ref. 47 and supporting information). Even models that are based on the same fundamental mechanism, such as duplication–mutation in DMC and DMR, are perfectly separable. Even small algorithmic changes in network mechanisms can thus give rise to easily detectable differences in substructures. Our results (see Fig. 1) confirm that although many of these models have similar degree distributions, clustering coefficients, or mean geodesic lengths, they have indeed distinguishable topologies.

Fig. 2 shows the first few decision nodes of a resulting ADT. The prediction scores reveal that a high count of 3-cycles suggests a DMC network (node 3). The DMC mechanism indeed facilitates the creation of many 3-cycles by allowing two copies to attach to each other, thus creating 3-cycles with their common neighbors. In particular a few combinations are good predictors for some classes. For example, a low count in 3-cycles combined with a high count in 8-edge linear chains is a good predictor for LPA and DMR networks (nodes 3 and 4). Because of the sparseness of the networks preferential attachment does not lead to a clustered structure. While LPA readily yields hubs, cycles are less probable. (Larger ADTs can be viewed in the supporting information.)

Having built a classifier enjoying good prediction accuracy, we can now determine the network mechanism that best reproduces the *Drosophila* protein network (or in principle any network of the same size) by using the trained ADTs for classification. Table 2 gives the prediction scores of the *Drosophila* network for each of the seven classes, averaged over folds.

The DMC mechanism is the only class having a positive prediction score in every case. In particular, for $p^* = 0.65$ the DMC classification has a high score of 8.2 ± 1.0 for eight-step subgraphs and 8.6 ± 1.1 for subgraphs with up to seven edges. Also, the comparatively small standard deviations over different folds indicate robustness of the classification against data subsampling. While the high rankings of both duplication–mutation classes confirm our biological understanding of protein network evolution, our findings strongly support an evolution restricted by functional complementarity over an evolution that creates and deletes functions at random.

Notably, for $p^* = 0.65$ the RDG mechanism of random growth (edges are connected randomly between existing vertices) has a higher prediction score than the LPA or AGV growing graph mechanisms. Growth without any underlying mechanism other than chance therefore generates networks closer in topology to the core network ($p^* = 0.65$) of *Drosophila* than growth governed by preferential attachment. We also emphasize that even though *Drosophila* exhibits the SMW *character* of high clustering and short mean geodesic length (34), the SMW *model* (4) (an interpolation between regular ring lattices and randomly connected graphs) does not accurately reproduce the *Drosophila* network. The classification for $p^* = 0.5$ is less confident, probably because of the additional noise present in the data when including low p value (improbable) interactions, as we discuss below.

Although not necessary for the classification itself, visualizing the distribution for each model and each subgraph, compared with that subgraph's census in *Drosophila,* can give a qualitative and more intuitive way of interpreting the classification result and a better understanding of the topological differences between *Drosophila* and each of the seven mechanisms. To this end we determine *rank scores* for every subgraph and mechanism, defined as the percentages of sampled networks that have a subgraph count above *Drosophila*'s count. A rank score of 50% corresponds to a distribution whose median is equal to *Drosophila*'s subgraph count. Fig. 4 shows the color-coded rank scores for every mechanism and every subgraph (only the subset of 51 subgraphs, which appear in the learned ADT, is shown here; see the supporting information for the full set). The subgraphs are ordered by similarity in rank scores (see caption of Fig. 4). A few subgraphs (S36–S51) featuring hubs without cycles are best modeled by the LPA mechanism; i.e., these subgraphs have rank scores close to 50%. For almost all other subgraphs, both duplication–mutation mechanisms (DMC and DMR) consistently have better rank scores than the other models. Notably, the SMW and RDS mechanisms have rank

Table 2. Prediction scores for the *Drosophila* protein network for different confidence thresholds *p and different cut-offs in subgraph size**

	Eight-step subgraphs ($p^* = 0.65$)		Subgraphs with up to seven edges ($p^* = 0.65$)		Eight-step subgraphs ($p^* = 0.5$)	
Rank	Class	Score	Class	Score	Class	Score
1	DMC	8.2 ± 1.0	DMC	8.6 ± 1.1	DMC	0.8 ± 2.9
2	DMR	−6.8 ± 0.9	DMR	−6.1 ± 1.7	DMR	−2.1 ± 2.0
3	RDG	−9.5 ± 2.3	RDG	−9.3 ± 1.6	AGV	−3.1 ± 2.2
4	AGV	−10.6 ± 4.2	AGV	−11.5 ± 4.1	LPA	−10.1 ± 3.1
5	LPA	−16.5 ± 3.4	LPA	−14.3 ± 3.2	SMW	−20.6 ± 1.9
6	SMW	−18.9 ± 0.7	SMW	−18.3 ± 1.9	RDS	−22.3 ± 1.7
7	RDS	−19.1 ± 2.3	RDS	−19.9 ± 1.5	RDG	−22.5 ± 4.7

Drosophila is consistently (independently of the cut-off in subgraph size) classified as a DMC network, with an especially strong prediction for a confidence threshold of $p^* = 0.65$.

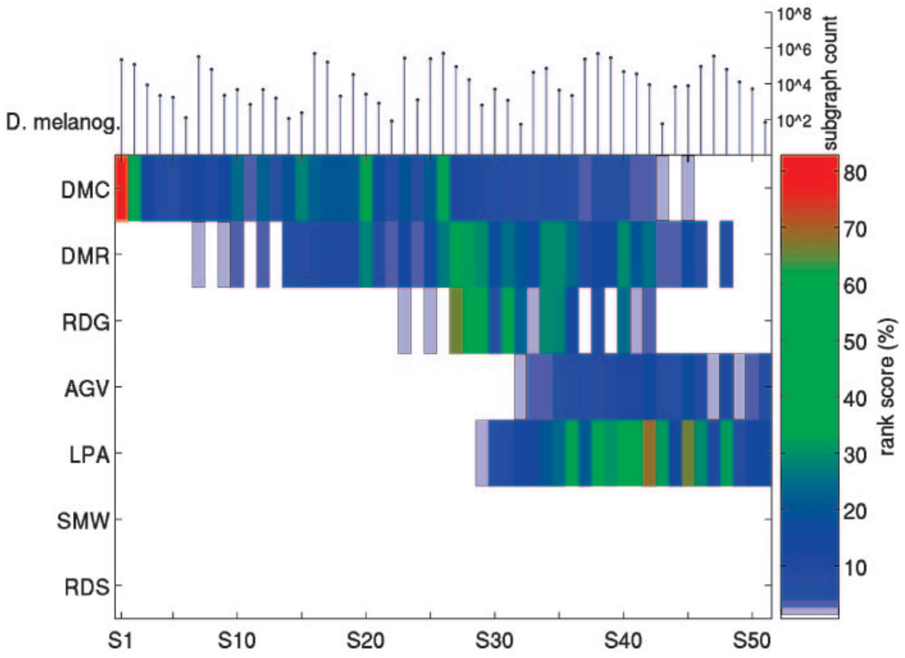


Fig. 4. Topological similarities and differences between *Drosophila* and each of the seven mechanisms. Color-coded rank scores are shown for a representative set of 51 subgraphs and every mechanism. The rank score $r_{i\alpha}$ for model *i* and subgraph α is defined as the percentage of sampled networks having a subgraph count above *Drosophila*'s value. The matrix of correlation coefficients $\rho_{\alpha\beta}$ of rank scores is then a similarity matrix between subgraphs. The coordinates of the eigenvector corresponding to the smallest nonzero eigenvalue [or Fiedler vector (50)] of the Laplacian (51) $L_{\alpha\beta} = \rho_{\alpha\beta} - \delta_{\alpha\beta}\Sigma_{\gamma}\,\rho_{\gamma\beta}$ (where $\delta_{\alpha\beta}$ is the Kronecker symbol, equal to 1 iff $\alpha = \beta$ can then be used to sort the subgraphs according to similarity in rank scores (for details see the supporting information). A rank score of 50% (green) corresponds to a distribution whose median is equal to *Drosophila*'s subgraph count. The labels S1–S51 refer to Fig. 3. The histogram in the upper part of the figure shows *Drosophila*'s subgraph counts.

scores 0 for all subgraphs; i.e., all sampled networks have lower subgraph counts than *Drosophila*. For a few subgraphs that feature long linear chains (S27–S33), the DMR model has better rank scores than DMC, whereas for almost all other subgraphs DMC has the best rank scores. In particular, DMC is the only model that can reach *Drosphila*'s counts for subgraphs S1–S26, which show complex cyclic structure.

Because yeast two-hybrid data are known to be susceptible to numerous errors (34), network analyses are reliable only if they are robust against noise. To confirm that our method shows this robustness, we classify the *Drosophila* network for various levels of artificially introduced noise by replacing existing edges with edges chosen at random. Fig. 5 shows the prediction scores for all seven classes as functions of the fraction of edges replaced. As validation, the network is correctly and confidently (p value $> 1 - 10^{-3}$) classified as an RDS graph when >80% of the edges are randomized. About 30% of *Drosophila*'s edges can be replaced without seeing any significant change in all seven prediction scores, and 40% can be replaced before *Drosophila* is no longer classified confidently as a DMC network. At this point the prediction scores of DMC, DMR, and AGV are very close, which is also observed for the prediction scores for $p^* = 0.5$ (see Table 2), where they rank top three in this order. The results therefore suggest that the less confident classification for $p^* = 0.5$ could be mainly due to the presence of more noise in the data after inclusion of low confidence-value edges.

We have presented a method to infer growth mechanisms for naturally occurring networks. Advantageous properties include robustness against both noise and data subsampling, and the absence of any prior assumptions about which network features are important. Moreover, because the learning algorithm does not assume any relationships among features, the input space can be generalized to include any additional statistics as potentially discriminative features. We find that the *Drosophila* protein interaction network is confidently classified as a DMC network, a result that strongly supports ideas presented by Vazquez *et al.* (6) and Force *et al.* (36) about the nature of genetic evolution,

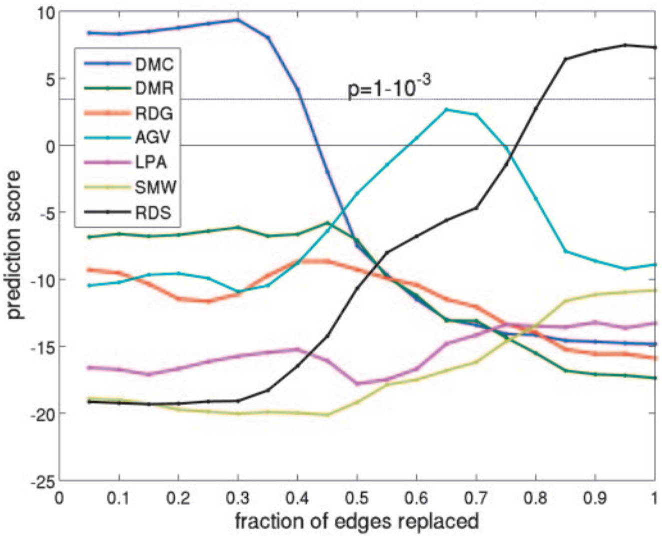


Fig. 5. Robustness against noise: Edges in *Drosophila* are randomly replaced, and the network is reclassified. Plotted are prediction scores for each of the seven classes as more and more edges are replaced. Each point is an average over 200 independent random replacements. For high noise levels (beyond 80%) the network is classified as an Erdös–Rényi (RDS) graph. Also note that the confidence in the classification as a DMC network for low noise (<30%) is even higher than in the classification as an RDS network for high noise. The prediction score $y(c)$ for class c is related to the estimated probability $p(c)$ for the tested network to be in class c by $p(c) = e^{2y(c)}/(1 + e^{2y(c)})$ (43). The dashed line indicates a p value of $1 - 10^{-3}$.

SEE COMMENTARY

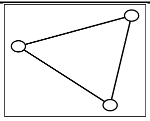
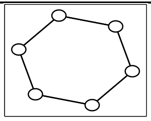
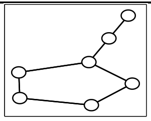
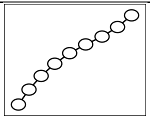
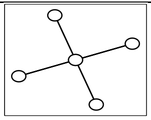
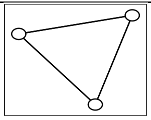
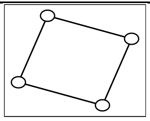
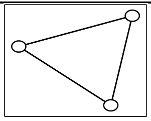
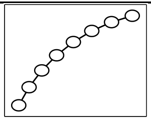
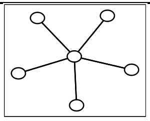
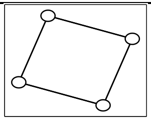
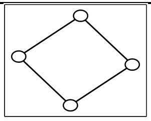
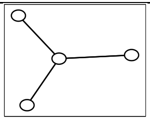
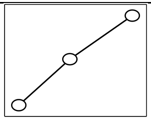
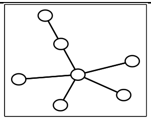
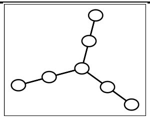
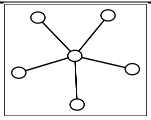
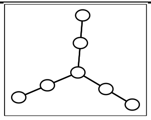
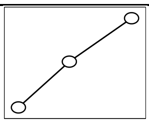
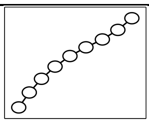
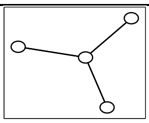
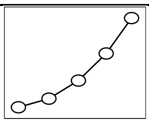
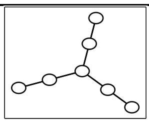
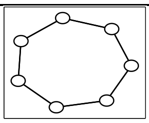
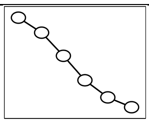
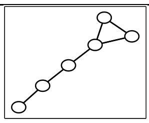
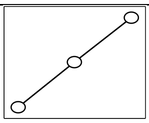
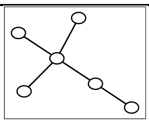
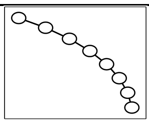
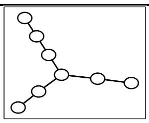
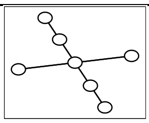
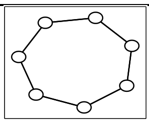
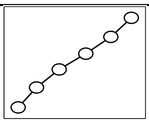
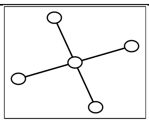
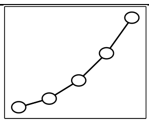
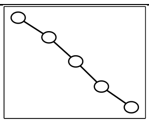
APPLIED MATHEMATICS

as well as recent direct experimental evidence presented by Wang *et al.* (48) for a single DMC event in *Drosophila melanogaster*. We also showed that different mechanisms, such as DMR, LPA, and RDG, model *Drosophila* well for different sets of subgraphs, a result which suggests that a model that mixes several mechanisms might be able to reproduce *Drosophila* even more accurately. Preliminary studies on the yeast protein–protein interaction network, as produced by an analysis that integrates multiple data sources (49), also strongly favors the DMC mechanism. We anticipate that further use of machine learning techniques will answer a number of such questions of interest in systems biology.

We acknowledge insightful discussions with Christina Leslie and Yoav Freund and key suggestions on the manuscript by G. Stolovitzky. This work was supported by National Science Foundation Grants ECS-0332479, ECS-0425850, and DMS-9810750 and National Institutes of Health Grant GM036277 (to C.H.W.).

1. Strogatz, S. H. (2001) *Nature* **410,** 268–276.
2. Newman, M. (2003) *SIAM Rev.* **45,** 167–256.
3. Barabási, A. (1999) *Science* **286,** 509–512.
4. Watts, D. & Strogatz, S. (1998) *Nature* **363,** 202–204.
5. de Solla Price, D. J. (1965) *Science* **149,** 510–515.
6. Vazquez, A., Flammini, A., Maritan, A. & Vespignani, A. (2003) *ComPlexUs* **1,** 38–44.
7. Sole, R. V., Pastor-Satorras, R., Smith, E. & Kepler, T. B. (2002) *Adv. Complex Syst.* **5,** 43–54.
8. Berg, J., Lässig, M. & Wagner, A. (2003) arXiv:cond-mat/0207711.
9. Rzhetsky, A. & Gomez, S. M. (2001) *Bioinformatics* **17,** 988–996.
10. Qian, J., Luscombe, N. M. & Gerstein, M. (2001) *J. Mol. Biol.* **313,** 673–681.
11. Bhan, A., Galas, D. J. & Dewey, T. G. (2002) *Bioinformatics* **18,** 1486–1493.
12. Kumar, R., Raghavan, P., Rajagopalan, S. & Sivakumar, D. (2000) in *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science*, ed. Blum, A. (Inst. Electrical Electronics Engineers, Piscataway, NJ), pp. 57–65.
13. Hastie, T., Tibshirani, R. & Friedman, J. (2001) *The Elements of Statistical Learning* (Springer, New York).
14. Devroye, L., Györfi, L. & Lugosi, G. (1996) *A Probabilistic Theory of Pattern Recognition* (Springer, New York).
15. Saito, R., Suzuki, H. & Hayashizaki, Y. (2003) *Bioinformatics* **19,** 756–763.
16. Goldberg, D. S & Roth, F. P. (2003) *Proc. Natl. Acad. Sci. USA* **100,** 4372–4376.
17. Morris, Q. D., Frey, B. J. & Paige, C. J. (2004) in *Advances in Neural Information Processing Systems 16*, eds. Thrun, S., Saul, L. K. & Schölkopf, B. (MIT Press, Cambridge, MA) pp. 385–393.
18. Gomez, S. M. & Rzhetsky, A. (2002) *Pac. Symp. Biocomput.*, 413–424.
19. Shen-Orr, S. S., Milo, R., Mangan, S. & Alon, U. (2002) *Nat. Genet.* **31,** 64–68.
20. Milo, R., Shen-Orr, S. S., Itzkovitz, S., Kashtan, N. & Alon, U. (2002) *Science* **298,** 824–827.
21. Artzy-Randrup, Y., Fleishman, S. J., Ben-Tal, N. & Stone, L. (2004) *Science* **305,** 1107.
22. Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R. & Alon, U. (2004) *Science* **305,** 1107.
23. Hasty, J., McMillen, D. & Collins, J. J. (2002) *Nature* **420,** 224–230.
24. Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., *et al.* (2002) *Science* **298,** 799–804.
25. Wuchty, S., Oltvai, Z. N. & Barabási, A.-L. (2003) *Nat. Genet.* **35,** 176–179.
26. Vespignani, A. (2003) *Nat. Genet.* **35,** 118–119.
27. Rosenfeld, N., Elowitz, M. & Alon, U. (2002) *J. Mol. Biol.* **323,** 785–793.
28. Mangan, S. & Alon, U. (2003) *Proc. Natl. Acad. Sci. USA* **100,** 11980–11985.
29. Ziv, E., Koytcheff, R. & Wiggins, C. H. (2003) arXiv:cond-mat/0306610.
30. Holland, P & Leinhardt, S. (1976) *Sociological Methodology* **7,** 1–45.
31. Wasserman, S., Faust, K. & Iacobucci, D. (1994) *Social Network Analysis: Methods and Applications* (Cambridge Univ. Press, Cambridge, U.K.).
32. Connor, E. F. & Simberloff, D. (1979) *Ecology* **60,** 1132–1140.
33. Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M. & Alon, U. (2004) *Science* **303,** 1538–1542.
34. Giot, L., Bader, J. S., Brouwer, C., Chaudhuri, A., Kuang, B., Li, Y., Hao, Y. L., Ooi, C. E., Godwin, B., Vitols, E., *et al.* (2003) *Science* **302,** 1727–1736.
35. Hughes, A. L. (1994) *Proc. R. Soc. London B* **256,** 119–124.
36. Force, A., Lynch, M., Pickett, F. B., Amores, A., Yan, Y.-L. & Postlethwait, J. (1999) *Genet. Soc. Am.* **151,** 1531–1545.
37. Erdös, P & Rényi, A. (1959) *Publicationes Mathematicae* **6,** 290–297.
38. Callaway, D., Hopcroft, J. E., Kleinberg, J. M., Newman, M. E. & Strogatz, S. H. (2001) *Phys. Rev. E.* **64,** 041902–041908.
39. Klemm, K. & Eguiluz, V. M. (2002) *Phys. Rev. E* **65,** 036123–036127.
40. Freund, Y. & Mason, L. (1999) in *Proceedings of the 16th International Conference on Machine Learning*, eds. Bratko, I. & Dzeroski, S. (Kaufmann, San Francisco, pp. 124–133.
41. Schapire, R. E. (2002) in *MSRI Workshop on Nonlinear Estimation and Classification*, eds. Denison, D. D., Hansen, M. H., Holmes, C. C., Mallick, B. & Yu, B. (Springer, New York), pp. 149–172.
42. Friedman, J., Hastie, T. & Tibshirani, R. (1998) *Ann. Stat.* **28,** 337–407.
43. Freund, Y. & Schapire, R. (1997) *J. Comput. Syst. Sci.* **55,** 119–139.
44. Schapire, R. E. & Singer, Y. (2000) *Machine Learning* **39,** 135–168.
45. Freund, Y & Schapire, R. (1999) *J. Jpn. Soc. Artif. Intell.* **14,** 711–780.
46. Middendorf, M., Kundaje, A., Wiggins, C., Freund, Y. & Leslie, C. (2004) *Bioinformatics* **20,** Suppl. 1, I232–I240.
47. Vazquez, A. (2003) *Phys. Rev. E* **67,** 056104–056118.
48. Wang, W., Yu, H. & Long, M. (2004) *Nat. Genet.* **36,** 523–527.
49. Troyanskaya, O. G., Dolinski, K., Owen, A. B., Altman, R. B. & Botstein, D. (2003) *Proc. Natl. Acad. Sci. USA* **100,** 8348–8353.
50. Fiedler, M. (1973) *Czech. Math. J.* **23,** 298–305.
51. Chung, F. R. K. (1997) *Spectral Graph Theory*, Regional Conference Series in Mathematics (Am. Math. Soc., Providence, RI).

Table 8. Most discriminative subgraphs: The twelve top-ranking subgraphs for every set of ADTs. Scores are average boosting iterations at which the subgraph appears in the ADT.

	8-step subgraphs $p^* = 0.65$		8-step subgraphs $p^* = 0.5$		subgraphs with up to 7 edges $p^* = 0.65$	
RANK	SCORE	SUBGRAPH	SCORE	SUBGRAPH	SCORE	SUBGRAPH
1	3		1		1	
2	4		2		3	
3	5		3		4	
4	6		4		5	
5	8.4		10		6	
6	9		14		7.6	
7	13.8		15.4		8	
8	15.6		21.2		10.2	
9	22.6		22.6		12.2	
10	30.8		31.4		12.6	
11	35.8		33.2		16.8	
12	38.6		36.4		20	

I. CONFIDENCE THRESHOLD

To exclude protein-protein interactions that are unlikely to occur *in vivo* we determine a confidence score threshold p^*, deciding whether or not the putative edge should be included in the analysis. We present results for two different choices: $p^* = 0.5$ as suggested in ref. 1 and $p^* = 0.65$ based on percolation. Fig. 6 shows the sizes of the ten biggest components of the protein network for all possible values of p^*. While lowering p^*, the network undergoes several percolation events in which two large components join together. The last major event occurs at a value of $p^* = 0.65$, where the network attains its major global structure. We also remove self-interactions because none of the proposed mechanisms allow for them. After eliminating isolated vertices the resulting networks consist of 3359 vertices and 2795 edges for $p^* = 0.65$, and 4625 vertices and 4683 edges for $p^* = 0.5$.

Table 3 shows the average size of the giant component for the networks in the training data, compared to the size of the giant component in *Drosophila*.

II. NETWORK GENERATION ALGORITHMS

We here present the pseudo-code for all seven network mechanisms investigated. Using these algorithms, we generate training data by sampling 1000 examples for each of the seven different network models with the same number of vertices N (not including isolated vertices) and the number of edges M as measured in *Drosophila*. Parameters not determined by the equal-size and equal-density constraints are sampled uniformly over a given interval. Because M and N do not explicitly appear as input parameters for some models, we allow for small intervals of $\pm 5\%$ around M and N, which in fact broaden the distributions, thus making the classification task even harder.

In this section, A designates the adjacency matrix of the network ($A_{ij} = 1$, iff i and j interact); N designates the final number of vertices; all random numbers r_i are drawn from a uniform distribution in $(0, 1)$; and $[x]_+$ and $[x]_-$ denote the next-larger (*ceil*) and next-smaller (*floor*) integer of a real number x, respectively.

A. Duplication-mutation preserving complementarity (DMC) (2)

parameters: $q_{del} \in [0,1], q_{con} \in [0,1]$

substrate: 2 vertices connected by a single edge.

```
for $i = 3 \dots N$
    choose a random vertex $n_0$ between 1 and $i - 1$
    for every neighbor $n$ of vertex $n_0$
        $A_{in} = 1;\ A_{ni} = 1;$
        draw random number $r_1$
        if $r_1 < q_{del}$
            draw random number $r_2$
            if $r_2 < \frac{1}{2}$
                $A_{nn_0} = 0;\ A_{n_0n} = 0;$
            else
                $A_{ni} = 0;\ A_{in} = 0;$
            end if
        end if
    end for
    draw random number $r_3$
    if $r_3 < q_{con}$
        $A_{in_0} = 1;\ A_{n_0i} = 1;$
    end if
end for
```

B. Duplication with random mutations (DMR) (3)

parameters: $q_{del} \in [0,1], q_{new} \in [0,1]$

substrate: 5-vertex-cycle

```
for $i = 6 \ldots N$
    $\alpha = \frac{q_{new}}{i-1}$
    choose random vertex $n_0$ between 1 and $i-1$
    for every neighbor $n$ of $n_0$
        $A_{ni} = 1;\ A_{in} = 1;$
        draw random number $r_1$
        if $r_1 < q_{del}$
            $A_{ni} = 0;\ A_{in} = 0;$
        end if
    end for
    for every vertex $j$ that is not a neighbor of $i$
        draw a random number $r_2$
        if $r_2 < \alpha$
            $A_{ij} = 1;\ A_{ji} = 1;$
        end if
    end for
end for
```

C. Random static networks (RDS) (4)

parameters: number of edges M

substrate: none

```
$N_e = 0$
while $N_e < M$
    choose random vertex $n_1$ between 1 and $N$
    choose random vertex $n_2$ between 1 and $N$
    if $A_{n_1 n_2} = 0$ & $n_1 \neq n_2$
        $A_{n_1 n_2} = 1;\ A_{n_2 n_1} = 1;$
        $N_e = N_e + 1;$
    end if
end while
```

D. Random growing networks (RDG) (5)

```
*parameters:* number of edges $M$
*substrate:* Static random graph of $i_0 = [2\frac{M}{N} + 1]_+$ vertices and $i_0\frac{M}{N}$ edges on average
$p = [\frac{M}{N}]_+ - \frac{M}{N};$
for $i = (i_0 + 1) \ldots N$
    $N_e = 0$
    draw random number $r_1$
    if $r_1 < p$
        $m_{tmp} = [\frac{M}{N}]_-;$
    else
        $m_{tmp} = [\frac{M}{N}]_+;$
    end if
    while $N_e < m_{tmp}$
        choose random vertex $n_1$ between 1 and $i$
        choose random vertex $n_2$ between 1 and $i$
        while $A_{n_1 n_2} = 1$ or $n_1 = n_2$
            choose random vertex $n_1$ between 1 and $i$
            choose random vertex $n_2$ between 1 and $i$
        end while
        $A_{n_1 n_2} = 1;\ A_{n_2 n_1} = 1;$
        $N_e = N_e + 1;$
    end while
end for
```

E. Linear preferential attachment networks (LPA) (6)

parameters: $a \in (0,5)$, number of edges M; (the upper limit of a was chosen after observing that the maximum degree in *Drosophila* is 35, and after noticing that the interesting behavior for preferential attachment networks arises for smaller a)

substrate: Static random graph of $i_0 = [2\frac{M}{N}+1]_+$ vertices and $i_0\frac{M}{N}$ edges on average

```
$p = [\frac{M}{N}]_+ - \frac{M}{N}$;
for $i = (i_0+1)\ldots N$
    $k_{j_1} = (\sum_{j_2} A_{j_1 j_2}) + a,$ for $j_1 = 1 \ldots i-1$;
    $N_e = 0$;
    $K_{j_1} = \sum_{j_2=1}^{j_1} k_{j_2},$ for $j_1 = 1 \ldots i-1$;
    draw random number $r_1$
    if $r_1 < p$
        $m_{tmp} = [\frac{M}{N}]_-$;
    else
        $m_{tmp} = [\frac{M}{N}]_+$;
    end if
    while $N_e < m_{tmp}$
        draw random number $r_2$;
        find $j$ such that $K_{j-1} < r_2 K_{i-1} < K_j$;
        if $A_{ij} = 0$
            $A_{ij} = 1;\ A_{ji} = 1$;
            $N_e = N_e + 1$;
        end if
    end while
end for
```

F. Aging vertices networks (AGV) (7)

parameters: $\mu \in (0,1)$, $a \in (0,1)$, number of edges M;

substrate: Static random graph of $i_0 = [2\frac{M}{N} + 1]_+$ vertices and $i_0\frac{M}{N}$ edges on average

```
$p = [\frac{M}{N}]_+ - \frac{M}{N}$;
choose $[\frac{M}{N}]_+$ active vertices at random;
$V =$ list of active vertex indices;
for $i = (i_0 + 1) \ldots N$
    $k_{j_1} = \sum_{j_2} A_{j_1 j_2}$, for $j_1 = 1 \ldots i-1$;
    $K_{j_1} = \sum_{j_2=1}^{j_1} k_{j_2}$, for $j_1 = 1 \ldots i-1$;
    draw random number $r_1$;
    if $r_1 < p$
        choose $v_0$ one of the active vertices in $V$ at random;
    else
        $v_0 = 0$
    end if
    for $j = V_1 \ldots V_m$
        if $j = v_0$
            continue with for loop;
        end if
        draw random number $r_2$
        if $r_2 > \mu$
            $A_{ij} = 1$; $A_{ji} = 1$;
        else
            $v = i$;
            while $v = i$ or $A_{vi} = 1$
                draw random number $r_3$;
                find $v$ such that $K_{v-1} < r_3 K_{i-1} < K_v$;
            end while
            $A_{iv} = 1$; $A_{vi} = 1$;
        end if
    end for
    $Q_i = 1/(K_{V_i} + a),$ for $i = 1 \dots m$;
    $Q^c_i = \sum_{j=1}^{i} Q_j,$ for $i = 1 \dots m$;
    draw random number $r_4$;
    find $j$ such that $Q^c_{j-1} < r_4 Q^c_m < Q^c_j$;
    remove $V_j$ from list $V$;
    add $i$ to list $V$;
end for
```

G. Small-world networks (SMW) (8)

parameters: $q_{rewire} \in (0, 1)$; number of edges M *substrate:* regular ring lattice of N vertices, where every vertex is connected to its neighbors at a maximum distance of $[\frac{M}{N}]_-$ and, with probability $[\frac{M}{N}]_+ - \frac{M}{N}$, also to a neighbor at distance $[\frac{M}{N}]_+$ such that the average total number of edges is M.

```
for all edges $(i, j)$ in random order
    draw random number $r_1$
    if $r_1 > q_{rewire}$
        continue with for loop;
    end if
    $V$ list of vertices $v$ such that $A_{iv} = 0$ & and $i \neq k$;
    draw random element $v$ of $V$;
    $A_{ij} = 0$; $A_{ji} = 0$;
    $A_{iv} = 1$; $A_{vi} = 1$;
end for
```

III. SUBGRAPH CENSUS

We emphasize the unbiased selection of topological substructures in our studies. We do not make prior assumptions about the importance of specific features; instead, we characterize the topological structure of a network by considering all subgraphs up to a given size. The cut-off in size can be chosen in different ways, such as in the number of edges, number of vertices, or the length (in steps) of a walk on the graph which serves to define the subgraph. To show that the classification results are robust against this choice we present results for two different cut-offs: *(i)* all subgraphs that can be constructed by a walk up to a length of eight edges (148 nonisomorphic subgraphs); and *(ii)* all subgraphs with a total number of edges up to seven, not necessarily constructed by a walk (130 nonisomorphic subgraphs). The two cases will be referred to as “8-step subgraphs” and “subgraphs with up to 7 edges”, respectively.

The first algorithm considers all possible walks in the network of length eight edges and groups them into isomorphism classes. At the end, the number of walks in every class is divided by a symmetry factor accounting for the fact that some subgraphs can be traversed by multiple distinct paths.

The second algorithm loops over all possible combinations of seven edges in the network, such that the set of edges represents a connected subgraph. It is possible to restrict the for-loops such that every subgraph in the network is encountered exactly once.

IV. ALTERNATING DECISION TREES

We exploit a learning algorithm that does not assume any relationships among features. Subgraph counts are usually not independent of each other, reflected in “conservation laws” (9) referring to the fact that some subgraphs contain others. Our analysis assumes neither dependence nor independence of the features. In particular, it does not embed the subgraph counts in a metric space, as for example in support vector machines (10). Features distinguish themselves solely by their individual ability to discriminate different classes. In particular, the classification results are independent of normalization. We are therefore able to combine different topological substructures without any further assumptions. Note that one can then consider features of completely different nature such as mean geodesic lengths, clustering coefficients, or power-law exponents (along with subgraph counts) in a single classifier.

An *Alternating Decision Tree* (ADT) (11) is a generalized decision tree that outputs a real-valued prediction score and that uses boosting (12) to learn the decision rules and scores. A trained ADT (Fig. 7) consists of alternating decision nodes (rectangles) and prediction nodes (ovals). A test network traces several paths in the tree by following all of the decision nodes emanating from a prediction node, but only a single prediction node ("true" or "false") emanating from a decision node. A prediction node contains scores for every class, which sum to the final prediction scores of the examples that reach the considered prediction node. The prediction for a given example is the class with the highest final prediction score. The absolute value of the score quantifies the confidence in the prediction (13).

ADTs are learned by adding a decision node (with its two prediction nodes) at each boosting iteration where the chosen decision rule minimizes the Adaboost loss function over all possible decision rules. A new decision node can be introduced at any previous prediction node; therefore several decision nodes can follow a prediction node.

In the one-vs-rest multi-class case used here, it is possible that for a given test network the prediction scores for all classes are negative, meaning that for every one-vs-rest combination, "rest" was always predicted. In this case, even though there is one class that comes closest to the tested network (the one with the highest prediction score), it should not be interpreted as the true class, but as the least erroneous. For example, the model might agree with some of the features but not with most of them. Different models might agree with the network of interest according to different sets of features.

For $p^* = 0.65$, we perform five-fold cross-validation (14) by partitioning the examples randomly into five sets (folds) of equal size. We then learn five ADTs, each of them trained on four out of five folds for 120 boosting iterations, and tested on the held-out fold. The performance of the classifier is measured in terms of the rate of misclassifications (0/1-loss), where we average test- and training-losses over folds. The test-loss then gives an empirical estimate of the generalization error, the probability of mislabeling an unseen test network (14). For $p^* = 0.5$, the networks are larger and less sparse, such that every mechanism gives rise to a broader distribution in the space of subgraph counts. We therefore use 20-fold cross-validation to reduce the fluctuations of the prediction scores over folds.

A. Learning algorithm

ADTs and their relationship with the Adaboost algorithm are explored in refs. 11-13 and 15-17. In the following we give a brief formal description of the learning algorithm for the one-vs-rest multi-class case.

Let π index the prediction nodes (ovals) in the ADT and let $P_{\pi,c}$ be the prediction score in node π for class c with $c \in \{1, \dots, N_c\}$, and where N_c is the number of existing classes (here, $N_c = 7$). Let $(\mathbf{x}_i, y_i)$ designate the training examples for $i \in \{1, \dots, N_E\}$, where N_E is the total number of training examples; let $\mathbf{x}_i$ designate the feature vector (subgraph counts), and y_i designate the class label ($y_i \in \{1 \dots N_c\}$). We write x_{ij} for the value of the feature j of example i. Every example i is also associated with N_c real-valued weights w_{ci} $(c = 1, \dots, N_c)$. Finally we define the sets $I_\pi = \{i | \mathbf{x}_i \text{ reaches } \pi\}$ of indices of examples that are able to reach the prediction node π. The w_{ci} are initialized to $w_{ci} = 1/N_E$ for all i and c.

The root prediction scores $P_{0,c}$ are given by

$$P_{0,c} = \frac{1}{2} \ln \frac{\sum_{\{i|y_i=c\}} w_{ci}}{\sum_{\{i|y_i \neq c\}} w_{ci}}.$$

Each boosting iteration introduces one decision node, together with its two prediction nodes. At each iteration, every possible new splitting node at every existing prediction node (indexed by σ) on every feature j and every possible split value ξ is tested; the combination accepted is that which minimizes the loss

$$\begin{aligned}L(\sigma, j, \xi) = \sum_{c=1}^{N_c} 2\sqrt{\sum_{\{i|i\in I_\sigma, x_{ij}<\xi, y_i=c\}} w_{ci} \sum_{\{i|i\in I_\sigma, x_{ij}<\xi, y_i\neq c\}} w_{ci}} \\ +2\sqrt{\sum_{\{i|i\in I_\sigma, x_{ij}\geq\xi, y_i=c\}} w_{ci} \sum_{\{i|i\in I_\sigma, x_{ij}\geq\xi, y_i\neq c\}} w_{ci}} \\ + \sum_{\{i|i\in\{1,\ldots,N_E\}, i\notin I_\sigma\}} w_{ci}.\end{aligned}$$

Let $(\sigma^*, j^*, \xi^*) = \operatorname{argmin}_{\sigma,j,\xi} L(\sigma, j, \xi)$; we then introduce a new decision node "$x_{ij^*} < \xi^*$", emanating from prediction node σ^*, associated with two prediction nodes π_1, π_2 with prediction scores

$$\begin{aligned}P_{\pi_1,c} &= \frac{1}{2}\ln\frac{\sum_{\{i|i\in I_{\sigma^*}, x_{ij}<\xi^*, y_i=c\}} w_{ci}}{\sum_{\{i|i\in I_{\sigma^*}, x_{ij}<\xi^*, y_i\neq c\}} w_{ci}} \\ P_{\pi_2,c} &= \frac{1}{2}\ln\frac{\sum_{\{i|i\in I_{\sigma^*}, x_{ij}\geq\xi^*, y_i=c\}} w_{ci}}{\sum_{\{i|i\in I_{\sigma^*}, x_{ij}\geq\xi^*, y_i\neq c\}} w_{ci}}.\end{aligned}$$

The boosting iteration closes by updating the weights $w_{ci} \leftarrow \exp(-F_c(\mathbf{x}_i)y_i)$ and normalizing them, where $F_c(\mathbf{x}_i)$ is the total prediction score of example i and class c given by summing over all prediction scores in the nodes that the example reaches in the current tree. This iterative construction of the prediction function $F_c(\mathbf{x}_i)$ may also be interpreted as additive logistic regression (18). Examples which are hard to predict therefore gain weight, and subsequent decision nodes concentrate on these misclassified examples, which is one of the main characteristics of the boosting algorithm (12) and large-margin classifiers such as support vector machines (19). We use the MLJAVA implementation of the ADT algorithm (16).

B. Most discriminative subgraphs

Using the trained ADTs, it is possible to rank subgraphs by discriminative strength, within the considered set of mechanisms. The earlier a feature is introduced into the ADT, the more discriminative it is (15). We can therefore rank features by earliest appearance in the ADTs, averaged over cross-validated folds. If a feature does not appear in a tree we assign to it an iteration index of $T + 1$ for this fold, where T is the total number of boosting iterations. Note that identifying most discriminative subgraphs is unrelated to "motif"-finding algorithms (20-28). The latter use randomizations of the original network to identify statistically significant subgraphs. Most discriminative subgraphs, identified using ADTs, are characterized by their ability to distinguish different network classes, and accordingly give structural information about the network mechanisms rather than the real data set.

V. CLASSIFICATION RESULTS

A. Prediction accuracy

Tables 4-6 show the probabilities $p(j|i)$ of predicting a class j given that the tested network is in class i. While DMC, DMR, RDG, and RDS have an almost perfect prediction accuracy, there are misclassifications among AGV, LPA, and SMW networks, since the AGV algorithm indeed has parameter regimes where it becomes equivalent to either LPA or SMW. The parameter $\mu \in (0, 1)$ controls the interpolation between a preferentially attached network and a network governed by active vertices. For $\mu \approx 1$, AGV is an LPA network. For $\mu \approx 0$, the sparsity forces the AGV network to become a regular ring lattice equivalent to a SMW network with $p_{rewire} \approx 0$. The misclassifications among those classes then are due to equivalent network mechanisms for specific parameter regimes. Yet the overall prediction accuracy is still 95-96% (Table 7) showing that network mechanisms do indeed give rise to very different topological substructures. Even models that are based on the same fundamental mechanism, like duplication-mutation in DMC and DMR, are almost perfectly separable. Only small algorithmic changes in network generation can thus give rise to easily detectable differences in substructures.

As illustrated above, some of the models can be described as a generalization of another model. One would expect to have a significant number of misclassifications for these models. However, our results show that subgraph space is sufficiently high-dimensional that such misclassifications are very rare. To build intuition, consider that the Erdös model (RDS) itself includes all possible network topologies. Nonetheless there is extremely low test loss with any other models.

B. Ranking of discriminative subgraphs

Table 8 shows the 12 most discriminative subgraphs, based on the iteration at which the subgraph is introduced, for all three classifiers. Interestingly, about 75% of the subgraphs belong to one of three groups (linear chains, hubs or cycles), whereas those groups form only about 10% of the whole input set of subgraphs. Thus the most discriminative subgraphs are, in general, those of highest symmetry. The 3-cycle appears in the top three subgraphs for all classifications, supporting the notion that the clustering coefficient is useful in distinguishing between different networks. Also note that the 4-cycle is a very discriminative subgraph. Linear chains with various lengths of nine, eight, six, and five vertices are among the top-ranking subgraphs, as well as hubs of degree three through five.

Closer inspection of the network-generating algorithms indicates why these particular groups of subgraphs are discriminative. “Rich-get-richer” (29) mechanisms such as LPA and AGV (for $\mu \approx 1$) readily yield hubs, whereas cycles are created when hubs are connected to each other, and are thus a second-order effect. In contrast, the duplication-mutation mechanisms DMC and DMR directly create cycles. When edges are copied, the twins and their common neighbors naturally form cyclic structures. Nevertheless, one might expect DMC and DMR to support hub structures, as duplication-mutation has been described as an *effective* preferential attachment (2).

C. Interpreting the ADT

Fig. 7 shows part of a learned ADT. Single nodes or paths in the tree with high prediction scores for specific mechanisms reveal subgraphs that are good discriminators. For example, a high count in triangles (S37) (see Fig. 10) is by itself a good predictor for DMC networks (nodes 3 and 43), which is mainly due to the existence of q_{con} (allowing twins to attach to each other, thus creating triangles with their common neighbors). In contrast, a low count in triangles combined with a high count in 8-edge linear chains (S134) is a good predictor for LPA networks (nodes 3 and 4). Linear chains occur frequently in LPA networks, as edges are concentrated on a small subset of vertices with high degrees, whereas cyclic structures are suppressed due to the networks' sparseness.

D. Classifying the *Drosophila* protein network

Table 9 shows the prediction scores of the *Drosophila* protein network for the three different classifiers. The DMC mechanism is the only class having a positive prediction score in every case. In particular, for $p^* = 0.65$ the DMC classification has a a high score of 8.2 ± 1.0 for 8-step subgraphs and of 8.6 ± 1.1 for subgraphs with up to 7 edges. Also, the comparatively small standard deviations over different folds indicate robustness of the classification against data subsampling. Note that the prediction scores for the first two columns are identical within fluctuations over folds, suggesting that the classification results are very robust against the nature of the cut-off in subgraph size. While the high rankings of both duplication-mutation classes confirm our biological understanding of protein network evolution, our findings strongly support an evolution restricted by functional complementarity over an evolution that creates and deletes functions at random.

Interestingly for $p^* = 0.65$ the RDG mechanism of random growth (edges are connected randomly between existing vertices) has a higher prediction score than the LPA or AGV growing graph mechanisms. Growth without any underlying mechanism other than chance therefore generates networks closer in topology to *Drosophila* than growth governed by preferential attachment. We also emphasize that the small-world *character* of high clustering and short mean geodesic length, often attributed to biological networks (1,30), is not enough to conclude that the given network is close to the small-world *model* (8) (an interpolation between regular ring lattices and randomly connected graphs), as shown here.

VI. VISUALIZING TOPOLOGICAL SIMILARITIES AND DIFFERENCES USING RANK SCORES

Visualizing the individual subgraph counts facilitates interpretation qualitatively and intuitively, though such a visualization is not necessary for classification. To render the topological differences between *Drosophila* and each of the seven mechanisms, we compare *rank scores* —– the percentage of network samples which for a given subgraph have a count higher than *Drosophila* (Figs. 9, 11, and 13). For a given subgraph and mechanism, a rank score of 50% corresponds to a distribution whose median is equal to *Drosophila*'s count for the considered subgraph. A rank score of 0% (100%) indicates that the considered mechanism consistently generates networks with a subgraph count lower (higher) than *Drosophila*'s count. In order to cluster subgraphs with similar rank scores $r_{i\alpha}$ over models, we compute the matrix of correlation coefficients $\rho_{\alpha\beta}$, where α, β range over subgraphs, and i ranges over network models. Entries in $\rho_{\alpha\beta}$ quantify the similarities between rank scores for different subgraphs. In spectral clustering (31), a general technique of computing an ordering from a similarity matrix is to consider its Laplacian $L_{\alpha\beta} = \rho_{\alpha\beta} - \delta_{\alpha\beta} \sum_{\gamma} \rho_{\gamma\beta}$. We here subtract the minimum element in ρ from all entries, since the Laplacian is usually associated with the diffusion operator on the graph, whose edges have nonnegative weights $\rho_{\alpha\beta}$. An ordering is then determined by sorting the coordinates of the eigenvector associated with the smallest nonzero eigenvalue of L [also called the Fiedler vector (31)].

For all three training data sets, DMC has the best overall rank scores; i.e. for almost all subgraphs, there are at least a few network samples that come close to *Drosophila*'s value. In particular, subgraphs with intricate cyclic structure (e.g., S2-S106 in Figs. 9 and 10, or S2-S68 in Figs. 11 and 12) are best modeled by DMC. By preserving complementary edges, the DMC algorithm is more inclined to produce structures with several interwoven cycles than DMR. The subgraphs for which DMR gives better rank scores (e.g., S107-S121 in Figs. 9 and 10, or S69-S102 in Figs. 11 and 12) feature 4-cycles, which often occur in *Drosophila* as well as in DMR networks. Moreover, *Drosophila* possesses many hubs (high-degree nodes), an attribute that is best modeled (in terms of rank scores) by the LPA mechanism (e.g., S134-S148 in Figs. 9 and 10, or S121-S148 in Figs. 11 and 12). For long linear chains (e.g., S122 and S128 in Figs. 9 and 10) RDG has the best rank scores. Thus *Drosophila* may be characterized by many cycles and high-degree nodes, which suggests a very clustered structure, despite its sparseness. (Similar observations can be made in Figs. 13 and 14.)

The RDS and SMW mechanisms have zero rank scores for almost all subgraphs and training sets. They both give rise to networks with many disconnected components and thus cannot generate hubs or cyclic structures. While the preferential attachement mechanisms LPA and AGV generate hubs comparable in number to those in *Drosophila*, only the duplication-mutation mechanisms, especially the DMC model, readily give rise to both cyclic structures *and* hubs.

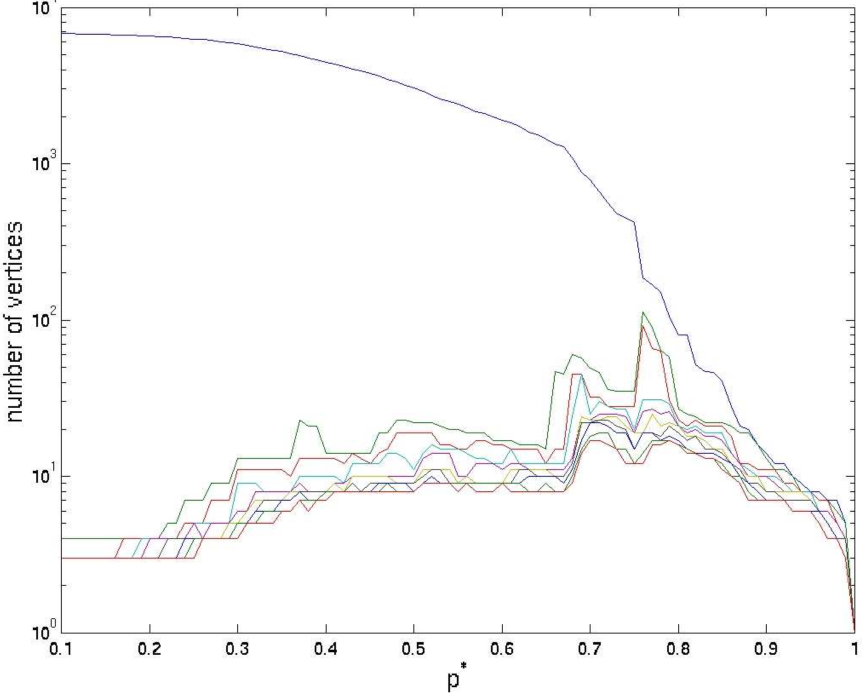
VII. ROBUSTNESS AGAINST NOISE

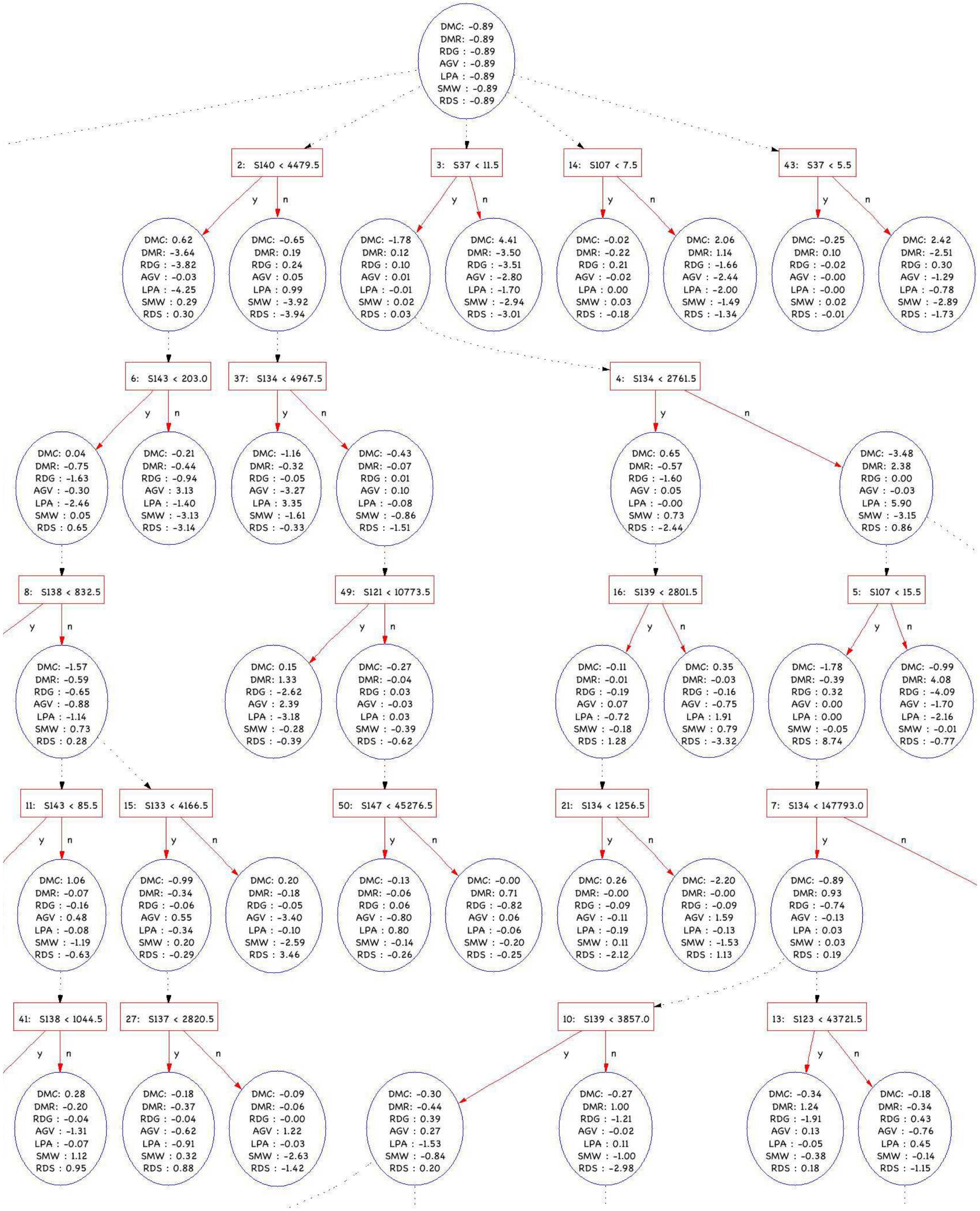
To test the robustness of *Drosophila*'s classification against noise, we artificially introduce noise into the *Drosophila* network by replacing existing edges by random ones, where we do not allow the creation of isolated vertices (as these are also removed in the training data for our classifier). We then classify the resulting networks for different noise levels using our learned ADTs. Fig. 5 shows the prediction scores for every class, averaged over 200 different realizations of the randomization procedure and averaged over the five cross-validated folds. About 45% of the edges can be replaced before the classification result of *Drosophila* changes from DMC to AGV. Fig. 15 gives a qualitative picture of how to interpret the change in prediction scores. The first 120 subgraphs mostly contain cyclic structures, whereas the last 28 subgraphs contain mostly hubs and linear chains. As the noise level is increased, all of these subgraph counts decrease, and *Drosophila*'s network loses more and more of its clustered structure. The cycles disappear around a noise level of 0.7, whereas a few hubs and linear chains still remain. This is the regime where the network is most similar to an AGV network, as shown by the prediction scores in Fig. 5. Eventually the *Drosophila* network is left only with a few hubs and linear chains, and the subgraph profile becomes indistinguishable from that of the RDS class in

Fig. 9.

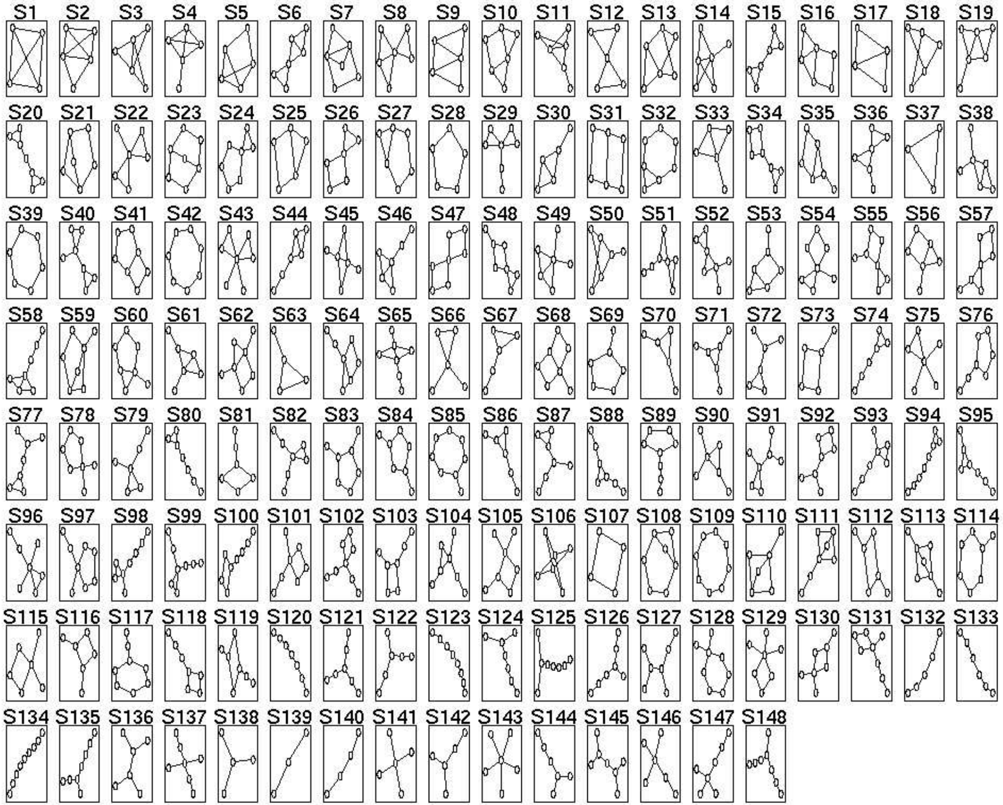
1. Giot, L., Lesage, G., Bader, G. D., Ding, H., Xu, H., Xin, X., Young, J., Berriz, G. F., Brost, R. L., Chang, M., *et al.* (2003) *Science* **302**, 1727–1736.
2. Vazquez, A., Flammini, A., Maritan, A. & Vespignani, A. (2003) *ComPlexUs* **1**, 38–44.
3. Sole, R. V., Pastor-Satorras, R., Smith, E. & Kepler, T. B. (2002) *Advances in Complex Systems* **5**, 43–54.
4. Erdös, P. & Rényi, A. (1959) *Publicationes Mathematicae* **6**, 290–297.
5. Callaway, D., Hopcroft, J. E., Kleinberg, J. M., Newman, M. E. & Strogatz, S. H. (2001) *Phys. Rev. E.* **64**, 041902, 1–7.
6. Barabási, A. (1999) *Science* **286**, 509–512.
7. Klemm, K & Eguiluz, V. M. (2002) *Phys. Rev. E* **65**, 036123, 1–5.
8. Watts, D. & Strogatz, S. (1998) *Nature* **363**, 202–204.
9. Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenshtat, I., Sheffer, M. & Alon, U. (2004) *Science* **303**, 1538–1542.
10. Jebara, T. (2003) *Machine Learning: Discriminative and Generative* (Kluwer Academic, Boston).
11. Freund, Y. & Mason, L. (1999) in *Proceedings of the 16th International Conference on Machine Learning,* eds. Bratko, I. & Dzeroski, S. (Morgan Kaufmann, San Francisco), 124–133.
12. Schapire, R. E. in *MSRI Workshop on nonlinear estimation and classification,* eds. Denison, D.D., Hansen, M.H., Holmes, C.C., Mallick, B. & Yu, B (Springer, New York), 149–172.
13. Schapire, R. E. & Singer, Y. (1999) *Machine Learning* **37**, 297–336.
14. Hastie, T., Tibshirani, R. & Friedman, J. (2001) *The Elements of Statistical Learning* (Springer, New York).
15. Middendorf, M., Kundaje, A., Wiggins, C., Freund, Y. & Leslie, C. (2004) *Bioinformatics* **20** Suppl. 1, I232–I240.
16. Schapire, R. E & Singer, Y. (2000) *Machine Learning* **39**, 135–168.

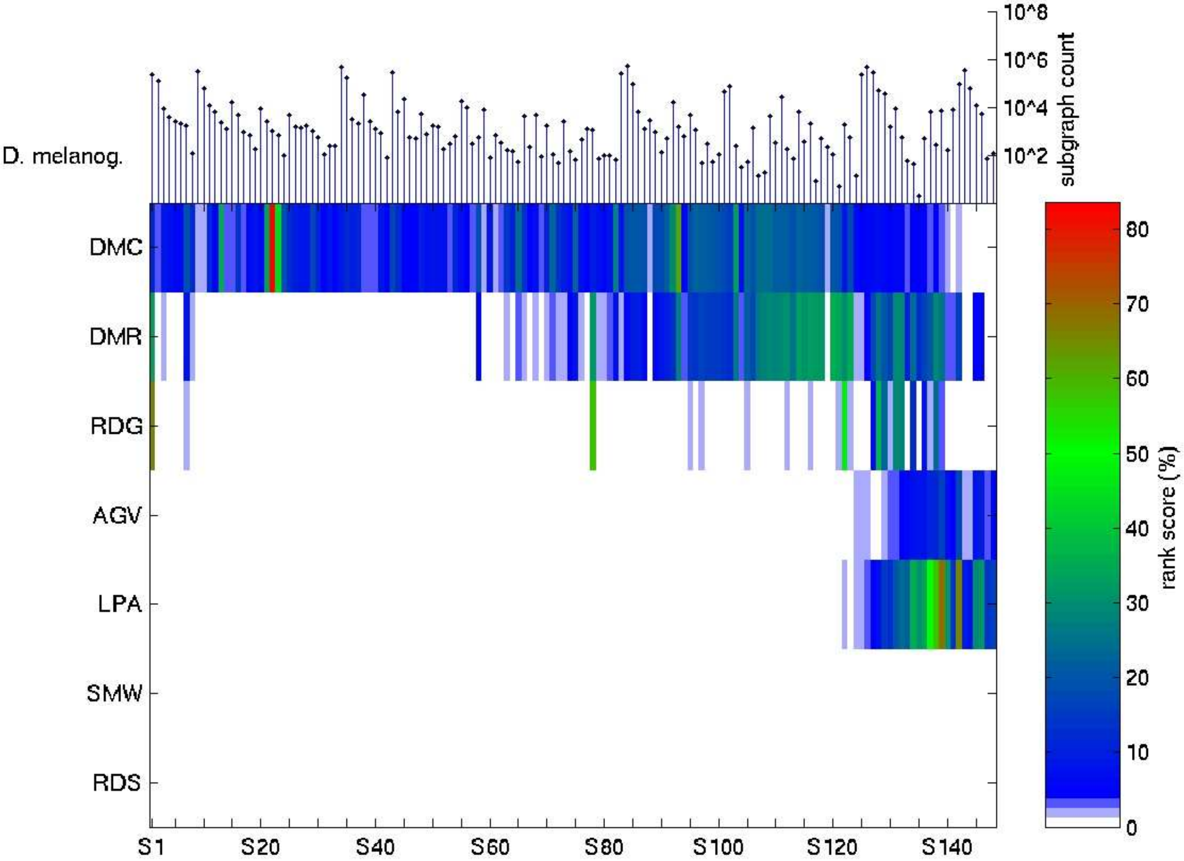
17. Middendorf, M., Kundaje, A., Wiggins, C., Freund, Y. & Leslie, C. (2005) in *RECOMB Satellite Workshop on Regulatory Genomics*, Lecture Notes in Bioinformatics, eds. Eskin, E. & Workman, C. (Springer, New York), 1–13.
18. Friedman, J., Hastie, T. & Tibshirani, R. (1998) *Ann. Statist.* **28**, 337–407.
19. Cristianini, N. & Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines*. (Cambridge, U.K.).
20. Shen-Orr, S. S., Milo, R., Mangan, S. & Alon, U. (2002) *Nat. Genetics* **31**, 64–68.
21. Milo, R., Shen-Orr, S. S., Itzkovitz, S., Kashtan, N. & Alon, U. (2002) *Science* **298**, 824–827.
22. Hasty, J., McMillen, D. & Collins, J. J. (2002) *Nature* **420**, 224–230.
23. Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., *et al.* (2002) *Science* **298**, 799–804.
24. Wuchty, S., Oltvai, Z. N. & Barabási, A.-L. (2003) *Nat. Genetics* **35**, 176–179.
25. Vespignani, A. (2003) *Nat. Genetics* **35**, 118–119.
26. Rosenfeld, N., Elowitz, M. & Alon, U. (2002) *J. Mol. Biol.* **323**, 785–793.
27. Mangan, S. & Alon, U. (2003) *Proc. Natl. Acad. Sci. USA* **100**, 11980–11985.
28. Ziv, E., Koytcheff, R. & Wiggins, C. H. (2003) arXiv:cond-mat/0306610.
29. Yule, G. (1925) *Philos. Trans. R. Soc. London B* **213**, 21–87.
30. Tong, A. H. Y., Bader, J. S., Brouwer, C., Chaudhuri, A., Kuang, B., Li, Y., Hao, Y. L., Ooi, C. E., Godwin, B., Vitols, E., *et al.* (2004) *Science* **303**, 808–813.
31. Chung, F. R. K. (1997) *Spectral Graph Theory*, Regional Conference Series in Mathematics (Am. Math. Soc., Providence, RI) No. 92.

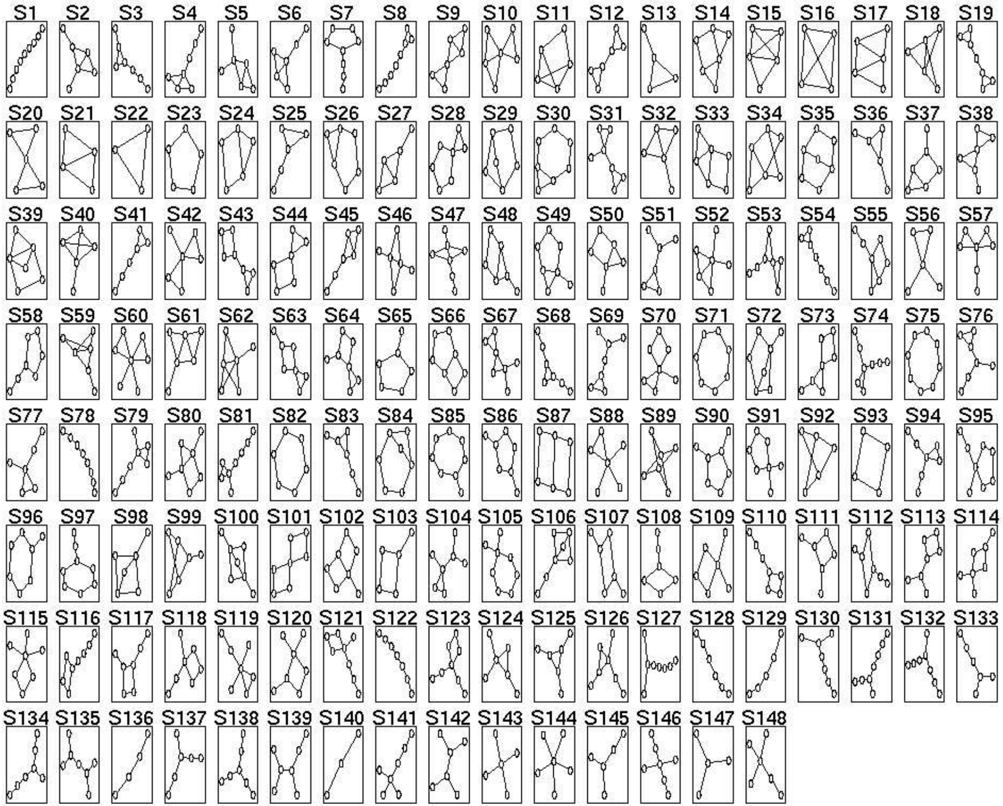

10^4
10^3
10^2
10^1
10^0
number of vertices
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
p^*

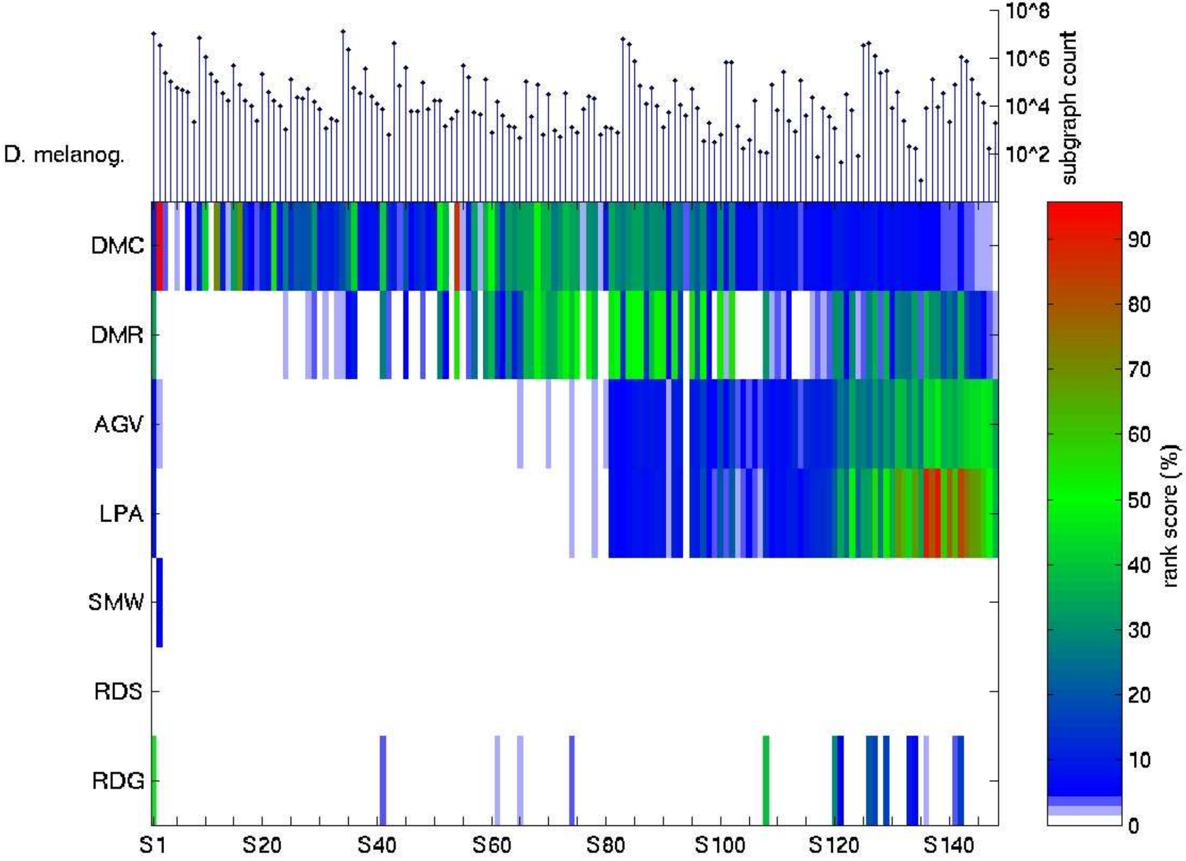


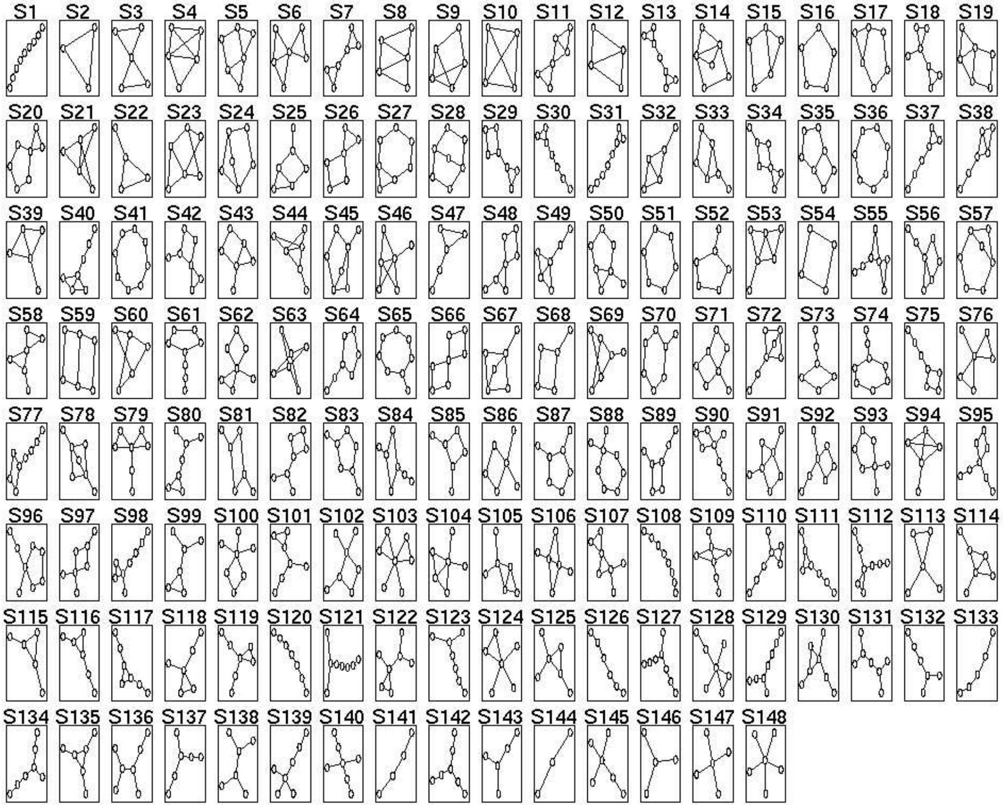
DMC: -0.89
DMR: -0.89
RDG : -0.89
AGV : -0.89
LPA : -0.89
SMW : -0.89
RDS : -0.89
2: S140 < 4479.5
y
n
3: S37 < 11.5
y
n
14: S107 < 7.5
y
n
43: S37 < 5.5
y
n
DMC: 0.62
DMR: -3.64
RDG : -3.82
AGV : -0.03
LPA : -4.25
SMW : 0.29
RDS : 0.30
DMC: -0.65
DMR: 0.19
RDG : 0.24
AGV : 0.05
LPA : 0.99
SMW : -3.92
RDS : -3.94
DMC: -1.78
DMR: 0.12
RDG : 0.10
AGV : 0.01
LPA : -0.01
SMW : 0.02
RDS : 0.03
DMC: 4.41
DMR: -3.50
RDG : -3.51
AGV : -2.80
LPA : -1.70
SMW : -2.94
RDS : -3.01
DMC: -0.02
DMR: -0.22
RDG : 0.21
AGV : -0.02
LPA : 0.00
SMW : 0.03
RDS : -0.18
DMC: 2.06
DMR: 1.14
RDG : -1.66
AGV : -2.44
LPA : -2.00
SMW : -1.49
RDS : -1.34
DMC: -0.25
DMR: 0.10
RDG : -0.02
AGV : -0.00
LPA : -0.00
SMW : 0.02
RDS : -0.01
DMC: 2.42
DMR: -2.51
RDG : 0.30
AGV : -1.29
LPA : -0.78
SMW : -2.89
RDS : -1.73
6: S143 < 203.0
y
n
37: S134 < 4967.5
y
n
4: S134 < 2761.5
y
n
DMC: 0.04
DMR: -0.75
RDG : -1.63
AGV : -0.30
LPA : -2.46
SMW : 0.05
RDS : 0.65
DMC: -0.21
DMR: -0.44
RDG : -0.94
AGV : 3.13
LPA : -1.40
SMW : -3.13
RDS : -3.14
DMC: -1.16
DMR: -0.32
RDG : -0.05
AGV : -3.27
LPA : 3.35
SMW : -1.61
RDS : -0.33
DMC: -0.43
DMR: -0.07
RDG : 0.01
AGV : 0.10
LPA : -0.08
SMW : -0.86
RDS : -1.51
DMC: 0.65
DMR: -0.57
RDG : -1.60
AGV : 0.05
LPA : -0.00
SMW : 0.73
RDS : -2.44
DMC: -3.48
DMR: 2.38
RDG : 0.00
AGV : -0.03
LPA : 5.90
SMW : -3.15
RDS : 0.86
8: S138 < 832.5
y
n
49: S121 < 10773.5
y
n
16: S139 < 2801.5
y
n
5: S107 < 15.5
y
n
DMC: -1.57
DMR: -0.59
RDG : -0.65
AGV : -0.88
LPA : -1.14
SMW : 0.73
RDS : 0.28
DMC: 0.15
DMR: 1.33
RDG : -2.62
AGV : 2.39
LPA : -3.18
SMW : -0.28
RDS : -0.39
DMC: -0.27
DMR: -0.04
RDG : 0.03
AGV : -0.03
LPA : 0.03
SMW : -0.39
RDS : -0.62
DMC: -0.11
DMR: -0.01
RDG : -0.19
AGV : 0.07
LPA : -0.72
SMW : -0.18
RDS : 1.28
DMC: 0.35
DMR: -0.03
RDG : -0.16
AGV : -0.75
LPA : 1.91
SMW : 0.79
RDS : -3.32
DMC: -1.78
DMR: -0.39
RDG : 0.32
AGV : 0.00
LPA : 0.00
SMW : -0.05
RDS : 8.74
DMC: -0.99
DMR: 4.08
RDG : -4.09
AGV : -1.70
LPA : -2.16
SMW : -0.01
RDS : -0.77
11: S143 < 85.5
y
n
15: S133 < 4166.5
y
n
50: S147 < 45276.5
y
n
21: S134 < 1256.5
y
n
7: S134 < 147793.0
y
n
DMC: 1.06
DMR: -0.07
RDG : -0.16
AGV : 0.48
LPA : -0.08
SMW : -1.19
RDS : -0.63
DMC: -0.99
DMR: -0.34
RDG : -0.06
AGV : 0.55
LPA : -0.34
SMW : 0.20
RDS : -0.29
DMC: 0.20
DMR: -0.18
RDG : -0.05
AGV : -3.40
LPA : -0.10
SMW : -2.59
RDS : 3.46
DMC: -0.13
DMR: -0.06
RDG : 0.06
AGV : -0.80
LPA : 0.80
SMW : -0.14
RDS : -0.26
DMC: -0.00
DMR: 0.71
RDG : -0.82
AGV : 0.06
LPA : -0.06
SMW : -0.20
RDS : -0.25
DMC: 0.26
DMR: -0.00
RDG : -0.09
AGV : -0.11
LPA : -0.19
SMW : 0.11
RDS : -2.12
DMC: -2.20
DMR: -0.00
RDG : -0.09
AGV : 1.59
LPA : -0.13
SMW : -1.53
RDS : 1.13
DMC: -0.89
DMR: 0.93
RDG : -0.74
AGV : -0.13
LPA : 0.03
SMW : 0.03
RDS : 0.19
41: S138 < 1044.5
y
n
27: S137 < 2820.5
y
n
10: S139 < 3857.0
y
n
13: S123 < 43721.5
y
n
DMC: 0.28
DMR: -0.20
RDG : -0.04
AGV : -1.31
LPA : -0.07
SMW : 1.12
RDS : 0.95
DMC: -0.18
DMR: -0.37
RDG : -0.04
AGV : -0.62
LPA : -0.91
SMW : 0.32
RDS : 0.88
DMC: -0.09
DMR: -0.06
RDG : -0.00
AGV : 1.22
LPA : -0.03
SMW : -2.63
RDS : -1.42
DMC: -0.30
DMR: -0.44
RDG : 0.39
AGV : 0.27
LPA : -1.53
SMW : -0.84
RDS : 0.20
DMC: -0.27
DMR: 1.00
RDG : -1.21
AGV : -0.02
LPA : 0.11
SMW : -1.00
RDS : -2.98
DMC: -0.34
DMR: 1.24
RDG : -1.91
AGV : 0.13
LPA : -0.05
SMW : -0.38
RDS : 0.18
DMC: -0.18
DMR: -0.34
RDG : 0.43
AGV : -0.76
LPA : 0.45
SMW : -0.14
RDS : -1.15

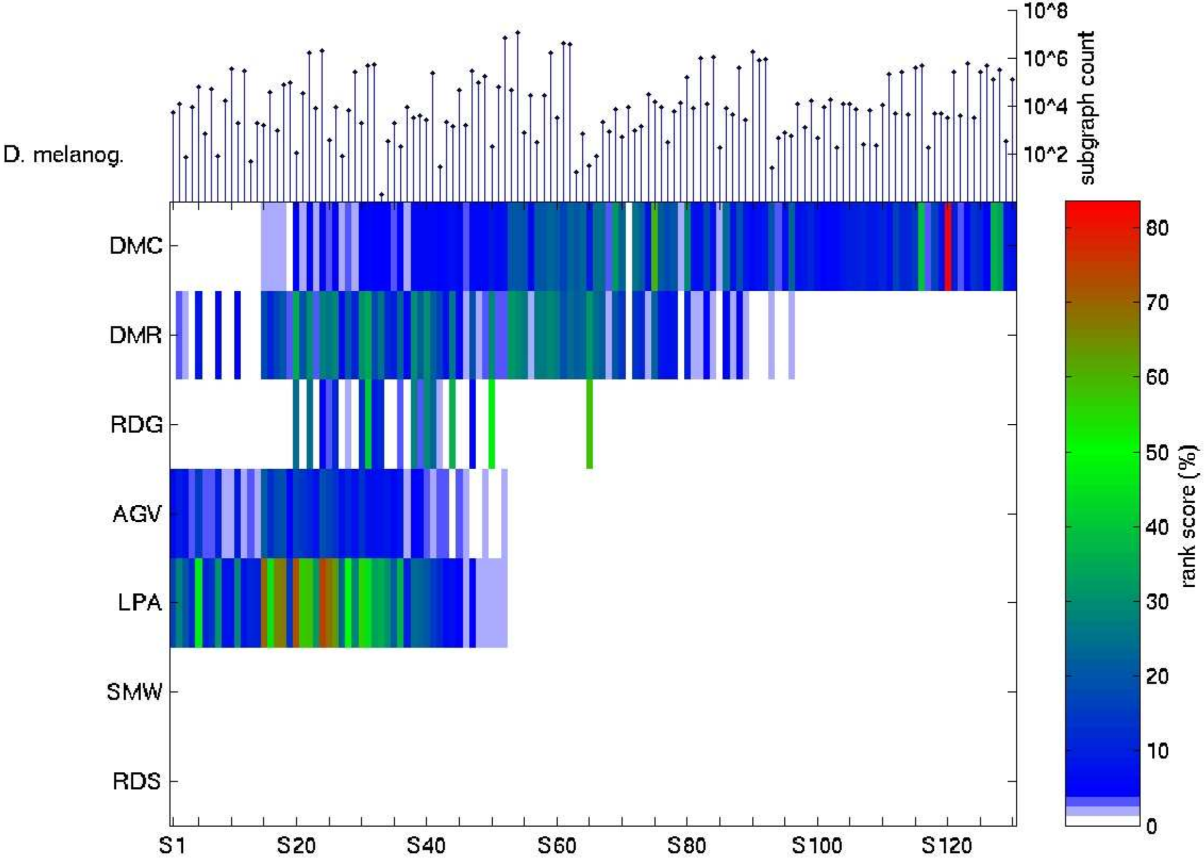

S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
S20
S21
S22
S23
S24
S25
S26
S27
S28
S29
S30
S31
S32
S33
S34
S35
S36
S37
S38
S39
S40
S41
S42
S43
S44
S45
S46
S47
S48
S49
S50
S51
S52
S53
S54
S55
S56
S57
S58
S59
S60
S61
S62
S63
S64
S65
S66
S67
S68
S69
S70
S71
S72
S73
S74
S75
S76
S77
S78
S79
S80
S81
S82
S83
S84
S85
S86
S87
S88
S89
S90
S91
S92
S93
S94
S95
S96
S97
S98
S99
S100
S101
S102
S103
S104
S105
S106
S107
S108
S109
S110
S111
S112
S113
S114
S115
S116
S117
S118
S119
S120
S121
S122
S123
S124
S125
S126
S127
S128
S129
S130
S131
S132
S133
S134
S135
S136
S137
S138
S139
S140
S141
S142
S143
S144
S145
S146
S147
S148

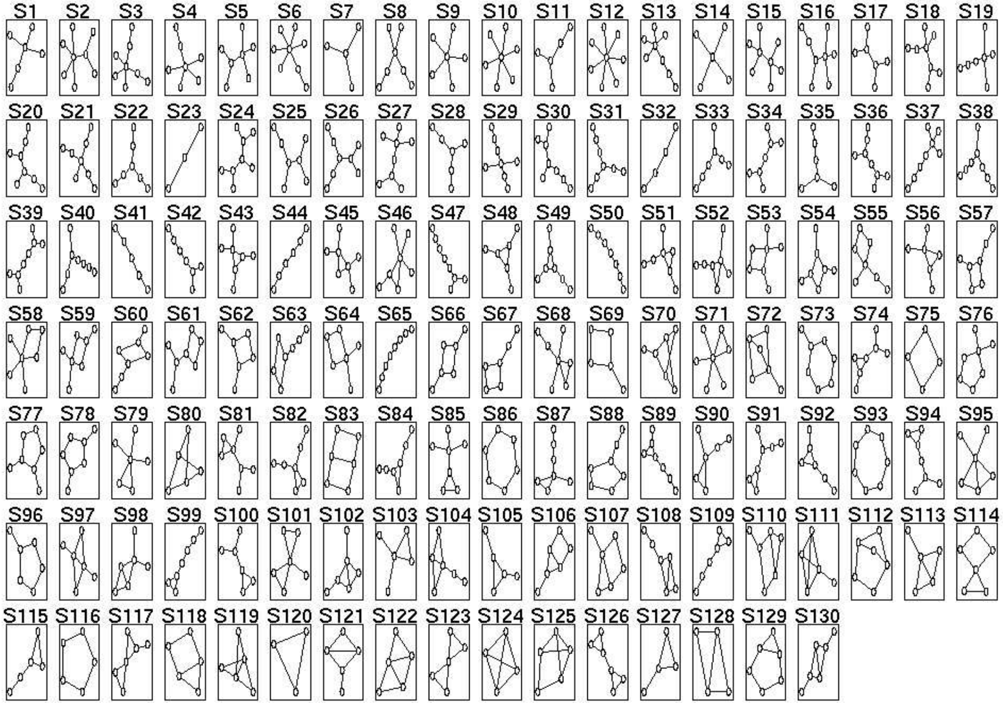

D. melanog.
DMC
DMR
RDG
AGV
LPA
SMW
RDS
S1
S20
S40
S60
S80
S100
S120
S140
10^8
10^6
10^4
10^2
subgraph count
rank score (%)
80
70
60
50
40
30
20
10
0

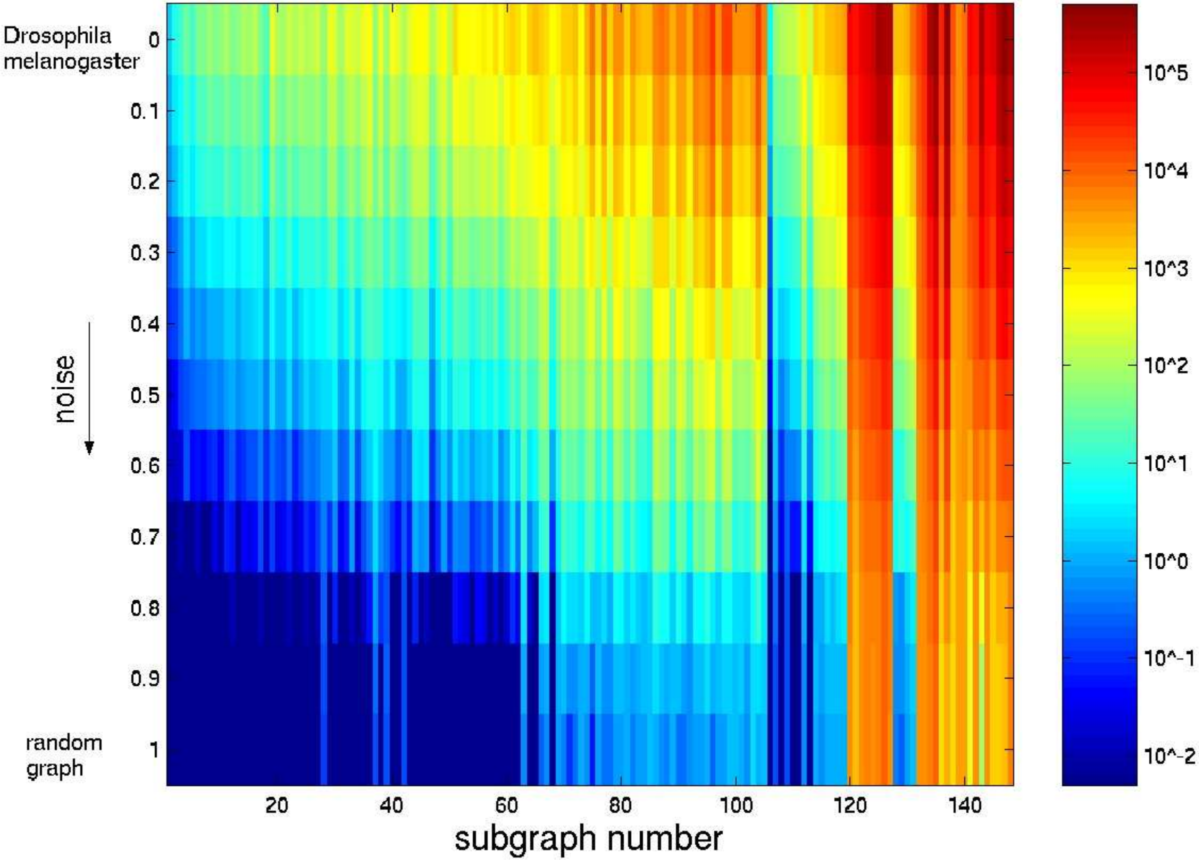

S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
S20
S21
S22
S23
S24
S25
S26
S27
S28
S29
S30
S31
S32
S33
S34
S35
S36
S37
S38
S39
S40
S41
S42
S43
S44
S45
S46
S47
S48
S49
S50
S51
S52
S53
S54
S55
S56
S57
S58
S59
S60
S61
S62
S63
S64
S65
S66
S67
S68
S69
S70
S71
S72
S73
S74
S75
S76
S77
S78
S79
S80
S81
S82
S83
S84
S85
S86
S87
S88
S89
S90
S91
S92
S93
S94
S95
S96
S97
S98
S99
S100
S101
S102
S103
S104
S105
S106
S107
S108
S109
S110
S111
S112
S113
S114
S115
S116
S117
S118
S119
S120
S121
S122
S123
S124
S125
S126
S127
S128
S129
S130
S131
S132
S133
S134
S135
S136
S137
S138
S139
S140
S141
S142
S143
S144
S145
S146
S147
S148


D. melanog.
10^8
10^6
10^4
10^2
subgraph count
DMC
DMR
AGV
LPA
SMW
RDS
RDG
S1
S20
S40
S60
S80
S100
S120
S140
90
80
70
60
50
40
30
20
10
0
rank score (%)


S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
S20
S21
S22
S23
S24
S25
S26
S27
S28
S29
S30
S31
S32
S33
S34
S35
S36
S37
S38
S39
S40
S41
S42
S43
S44
S45
S46
S47
S48
S49
S50
S51
S52
S53
S54
S55
S56
S57
S58
S59
S60
S61
S62
S63
S64
S65
S66
S67
S68
S69
S70
S71
S72
S73
S74
S75
S76
S77
S78
S79
S80
S81
S82
S83
S84
S85
S86
S87
S88
S89
S90
S91
S92
S93
S94
S95
S96
S97
S98
S99
S100
S101
S102
S103
S104
S105
S106
S107
S108
S109
S110
S111
S112
S113
S114
S115
S116
S117
S118
S119
S120
S121
S122
S123
S124
S125
S126
S127
S128
S129
S130
S131
S132
S133
S134
S135
S136
S137
S138
S139
S140
S141
S142
S143
S144
S145
S146
S147
S148


D. melanog.
10^8
10^6
10^4
10^2
subgraph count
DMC
DMR
RDG
AGV
LPA
SMW
RDS
S1
S20
S40
S60
S80
S100
S120
80
70
60
50
40
30
20
10
0
rank score (%)


S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
S14
S15
S16
S17
S18
S19
S20
S21
S22
S23
S24
S25
S26
S27
S28
S29
S30
S31
S32
S33
S34
S35
S36
S37
S38
S39
S40
S41
S42
S43
S44
S45
S46
S47
S48
S49
S50
S51
S52
S53
S54
S55
S56
S57
S58
S59
S60
S61
S62
S63
S64
S65
S66
S67
S68
S69
S70
S71
S72
S73
S74
S75
S76
S77
S78
S79
S80
S81
S82
S83
S84
S85
S86
S87
S88
S89
S90
S91
S92
S93
S94
S95
S96
S97
S98
S99
S100
S101
S102
S103
S104
S105
S106
S107
S108
S109
S110
S111
S112
S113
S114
S115
S116
S117
S118
S119
S120
S121
S122
S123
S124
S125
S126
S127
S128
S129
S130


Drosophila melanogaster
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
random graph
1
noise
20
40
60
80
100
120
140
subgraph number
10^5
10^4
10^3
10^2
10^1
10^0
10^-1
10^-2