# Supporting Information

Ahmed and Xing 10.1073/pnas.0901910106

## SI Text

**Related Work.** In this article, we addressed the problem of recovering the latent structure of time-varying networks from time series of nodal states. In this section, we draw connections to relevant contributions in both the fields of network analysis, structure recovery, and spatiotemporal regression.

The field of network analysis is concerned with analyzing networks data that describe the interconnection patterns between a set of actors in the network. Most of the work in network analysis focuses on the case where the network structure is observed, and seeks to characterize and model signature graph statistics or patterns in such networks. For time-invariant networks, represented as a single directed or undirected graph, a number of flexible statistical models have been proposed, including the classic exponential random graph models (ERGM) and extensions (1–3), latent space models that aim toward clustering and community discovery (4), and mixed-membership block models for role discovery (5). In the dynamic setting, where a time series of observations of the network structure is available, several formalisms have been proposed to model the dynamics of topological changes of such networks over time, including the continuous-time Markov process models (6) and the discrete-time models (7–9). This progress notwithstanding, the above techniques rely heavily on the availability of direct observations of the network structures, which can be difficult or impossible to obtain in reality, especially in a dynamic context, as discussed in the introduction. In this article, our focus is on recovering the latent time-varying networks rather than discovering and modeling patterns in observed networks.

Our proposed method stems from and builds on recent surge of interest in machine learning and statistical inference of the structures in high-dimensional multivariate functions, in particular, structural estimation of graphical models over static networks of actors based on samples of actors' (nodal) states. Earlier approaches view structure learning as a discrete combinatorial search problem and employ heuristic search methods (10, 11). Recent trends treat the structure learning problem as a model selection (also known as neighborhood selection) problem under loss functions motivated by a regularized regression between nodal states. These approaches build on the key insight that an $l_1$-regularization penalty over regression/classification coefficients can derive the coefficients corresponding to irreverent nodes to zero, whereby a sparse but consistent estimate of the structure can be recovered via solving a convex optimization problem (12–15). The specific form of the optimization problem depends on whether the nodal states are assumed to be continuous (13, 16–19) or discrete (20, 21).

However, with a few exceptions (22, 23), little has been done toward developing efficient learning techniques in dynamic contexts for recovering latent network topologies from observed attributes of entities constituting the network. The work presented in this article took a step toward bridging this gap.

Because TESLA inherently finds a temporally smoothed and regularized solution to a time-varying regression problem, perhaps it is beneficial to differentiate our work from contributions in the areas of spatial, temporal, and spatiotemporal regression that found a widespread use in ecology (24), social science (25), econometrics (26), and survival analysis (27). The input to such models is a known dependency network between a set of nodes of interest. These nodes each represent an entity being analyzed: for example, an individual, a city, a country, etc. Each of these nodes is associated with a regression problem on a shared covariate space; in other words, each of these nodes corresponds to a response variable (the output of the regression problem), along with a set of common covariates (the input to the regression problem), and the goal of the model is to investigate the effect of the dependency structure on the regression problem at each node. Therefore, these models in general assume a fixed structure over the nodes in the network in contrast to TESLA, which aims toward recovering such structure. In temporal regression, the goal is to allow the magnitude of the regression coefficients to vary over time to account for the change in the contribution of each covariate to the response variable (27). Therefore, these models differ from TESLA's formulation of which the goal is to delineate the time period in which each covariate is actually relevant to the response variable, which allows for structural recovery of the time-varying network, as opposed to merely the change of contribution of each covariate over time, which does not facilitate structural recovery.

**Hyperparameters Selection.** The hyperparameters in Eq. **2** trade off spareness, smoothness versus locally fitting the time-epoch specific samples. Larger values of $\lambda_2$ degenerate the problem into the static case in which all of the time-specific parameters are equal to each other, while setting $\lambda_2$ to zero decouples Eq. **2** into a set of independent $T$ $l_1$-regularized logistic regression problems, one at each epoch. Intuitively, the optimal value of $\lambda_2$ depends on both the evolution rate of the network and the number of samples available at each epoch. A fast-evolving network calls for lower values of $\lambda_2$, whereas a small number of samples at each epoch calls for slightly larger values for $\lambda_2$ that would tightly integrate all of the available samples in recovering the time-varying network. On the other hand, $\lambda_1$ controls the spareness of the resulting networks. For instance, setting $\lambda_1$ to zero results in a complete network, and as $\lambda_1$ increases, fewer edges are recovered until the empty network is reached for larger values of $\lambda_1$. Thus, small values of $\lambda_1$ favor recall, whereas larger values favor precision of the recovered edges. Moreover, these regularization parameters can be used as knobs to be tuned manually according to the data analyzer's goal. To tune these parameters automatically, we note that Eq. **2** can be regarded as a supervised classification problem, and thus many techniques can be used to select $(\lambda_1, \lambda_2)$ among different candidates. When there are enough data, cross-validation, or held-out datasets can be used; alternatively, the BIC score can be used. We define the BIC score for $(\boldsymbol{\theta}_i^1, \ldots, \boldsymbol{\theta}_i^T)$ to be:

$$\mathrm{BIC}(\boldsymbol{\theta}_i^1, \ldots, \boldsymbol{\theta}_i^T) \approx \sum_{t=1}^{T} \ell(\mathbf{x}^t; \boldsymbol{\theta}_i^t) - \frac{\log(\sum_{t=1}^{T} N_t)}{2} \mathrm{Dim}(\boldsymbol{\theta}_i^1, \ldots, \boldsymbol{\theta}_i^T), \quad \textbf{[s1]}$$

where $\mathrm{Dim}(\cdot)$ denotes the dimensionality of the estimated values. Similar to Tibshirani et al. (30), we adopt the following definition, which counts the number of runs of nonzero parameter values.

$$\mathrm{Dim}(\boldsymbol{\theta}_i^1, \ldots, \boldsymbol{\theta}_i^T) = \sum_{t=1}^{T} \sum_{j=1}^{P} I[\theta_{ij}^t \neq 0 \text{ and } \theta_{ij}^{t-1} = 0]. \quad \textbf{[s2]}$$

Note that the above definition is lenient to perturbations to the values of the actual edge weights (i.e., it counts the total number of edges), whereas, the definition in ref. 30 strictly increases the dimensionality of the model with these perturbations. Our intuition here is that a nonzero parameter corresponds to an edge, thus the above measure counts the total number of edges.

**Generation of Simulation Data.** In this section we provide more details of the scheme used to generate the synthetic time-varying network. The synthetic time-varying network is parameterized by a 3-tuple $(p, d, r)$, where $p$ is the number of nodes in the network, $d$ is the average degree of nodes in the first network, and $r$ is the rate of change modeled as the number of added and deleted edges at each change point. The generative process proceeds as follows. We begin by generating the network at the first epoch by selecting for each node a set of maximum $d$ neighboring nodes uniformly at random. We proceed from node 1 up to node $p$. To generate the neighborhood for node $i$ that already has $n_i$ incident edges to previously processed nodes, we select uniformly at random $\max(0, d - n_i)$ nodes from the set $\{j \mid i < j \leq p \text{ and } n_j < d\}$, and add edges between them and node $i$. Once the first network is generated, we sample a weight for each generated edge uniformly at random from the interval $[-1,-0.5] \cup [0.5,1]$. To generate the time-varying networks of subsequent time points, for every 10 epochs, we add $r$ edges and delete $r$ edges from the set of existing edges. The probability of adding an edge to a given node is inversely proportional to the number of edges already added to or deleted from this node. Moreover, to avoid unnecessary fragmentation of the lifespans of the edges in the network, we set the probability of deleting an existing edge to be proportional to its current lifetime (i.e., the duration between the current epoch and the epoch in which this edge was born). Once an edge is added, its weight is sampled form the interval $[-1,-0.5] \cup [0.5,1]$. We run this process consecutively for 10 times; thus, the number of total epochs is $T = 100$, and the overall result of this process is a set of $T$ MRFs: one at each epoch. Finally, we generate a set of $N_t$ samples of nodes' states at each epoch using Gibbs sampling with a 1,000 burn-in iterations and a lag of 200 iterations between collected samples. These time-stamped samples constitute the input to the static algorithm and TESLA. In our experiments, we fixed the number of nodes in the network, $p$, at 50 nodes.

**Network Visualization.** To visualize the recovered time-varying networks, we used an array of different layout strategies. Different layouts are tailored for different types of networks and for different visualization purposes. For the biological network, we used a circular layout with fixed ordering over the nodes, which is one of the standard layouts for visualizing gene networks. For the Senate data set, we used the organic layout as implemented in the Sytoscape package (www.cytoscape.org). The organic layout is tailored for undirected graphs, and is based on the force-directed layout paradigm (31). This layout produces a representation of networks that often exposes the inherent symmetric and clustered structure of a graph, as well as a well-balanced distribution of nodes and few edge crossings (32). These characteristics make it a perfect choice to illustrate how well TESLA recovers the political party structure in the Senate dataset. For the NIPS dataset, we used the radial layout (33) as implemented in the GraphViz software package (www.graphviz.org). For this layout, one node, the focus node, is chosen automatically as the center and put at the origin. The remaining nodes are placed on a sequence of concentric circles centered around the origin, each with a fixed radial distance from the previous circle. The process is repeated for every connected component of the graph that might result in different focus nodes. This layout emphasizes the relationship between nodes in the graph and the focus node(s), which makes it a suitable choice to track the evolution of the neighborhood of selected keywords and authors in the NIPS dataset.

**Evolving Gene Networks During *Drosophila melanogaster* Development.** ***Preprocessing the gene networks.*** We used the microarray time series data collected by Arbeitman et al. (34) in their study of the gene expression patterns during the life cycle of *D. melanogaster*. Approximately 9,700 *Drosophila* cDNA elements representing 5,081 different genes were used to construct the 2-color spotted cDNA microarrays. The genes analyzed in this article consist of a subset of 4,028 sequence-verified, unique genes. Experimental samples were measured at 66 different time points spanning the embryonic, larval, pupal, and adulthood periods. Each hybridization is a comparison of one sample to a common reference sample made from pooled mRNA representing all stages of the life cycle. Normalization is performed so that the dye-dependent intensive response is removed, and the average ratio of signals from the experimental and reference sample equals one. The final expression value is the log ratio of signals. Missing values are imputed in the same manner as in Zhao et al. (35). This is based on the assumption that gene expression values change smoothly over time. If there is a missing value, a simple linear interpolation using values from adjacent time points is used, i.e., the value of the missed time point is set to the mean of its 2 neighbors. When the missing point is a start or end point, it is simply filled with the value of its nearest neighbor. The expression values are quantitized into binary numbers using thresholds specific to each gene in the same manner as in ref. 35. For each gene, the expression values are first sorted; then the top 2 extreme values in either end of the sorted list are discarded; last, the median of the remaining values is used as the threshold above which the value is binarized as 1 and 0 otherwise. Here, 1 means the expression of a gene is up-regulated, and 0 means down-regulated.

***A "movie" of the* Drosophila *development gene network aging.*** In Fig. S1, we show a "movie" of the *D. melanogaster* developmental gene network as the organism ages, where genes are ordered according to their top level biological function. The dynamic networks appear to rewire over time in response to the developmental requirement of the organism. For instance, in the middle of embryonic stage (time point 4), most of genes selectively interact with other genes which results in a sparse network consisting mainly of paths. In contrast, near the end of the adulthood stage (time point 23), genes are more active, and each gene interacts with many other genes, which leads to visible clusters of gene interactions.

***Dynamics of known gene interactions.*** Different gene interactions may follow distinctive temporal programs of activation, appearing and disappearing at different time points during the life cycle of the *D. melanogaster*. In turn, the transient nature of the interactions implies that the evidence supporting the presence of an interaction between 2 genes may not be present in all microarray experiments conducted during different developmental stages of the organism. Therefore, pooling all microarray measurements and inferring a single static network can undermine the inference process rather than helping it. This problem can be overcome by learning a time-varying network that recovers transient interactions that are supported by a subset of the experiments.

To show the advantage of recovering a dynamic network over a static network, the recovered interactions (i.e., edges) by these 2 types of networks are compared against a list of known undirected gene interactions recorded in Flybase. TESLA recovers 50% more known gene interactions than the gene interactions in the static network. Furthermore, the static network provides no information on when a gene interaction starts or ends. In contrast, TESLA pinpoints the temporal on-and-off sequence for each recovered gene interaction. In Fig. S2, we

visualize the onset of the activation patterns of known gene interactions recorded in Flybase. To ease visualization, hierarchical clustering is performed on these sets of recovered gene interactions based on their activation patterns. It can be seen that all these interactions are transient and very specific to a certain stage of the life cycle of the *D. melanogaster*.

**Transient network hubs.** We also examined the recovered networks for hub transcription factor (TF) nodes. Hubs are high-degree nodes in the networks. They may represent the most influential elements of a network and tend to be essential for the developmental processes of the organism. The top 20 hubs are identified and tracked over time in terms of their degrees, and this evolution is visualized as a color map in Fig. S2*A*. The degrees of the transcriptional factors peak at different stages, which suggests that they may differentially trigger targeted genes based on the biological requirements of the developmental processes. In Fig. S2 *B* and *C*, a functional decomposition and enrichment analysis is performed on the target genes regulated by 2 example transcriptional factor hubs. For instance, *peb*, the protein ejaculatory bulb, mainly interacts with extracellular region genes and genes involved in structural molecular activity. Another example is the *spt4*, which seems to trigger many binding genes. This is consistent with its functional role in chromatin binding and zinc ion binding.

Again, because this microarray dataset covers only a limited fraction of all of the *Drosophila* genes and misses many of the important *Drosophila* developmental genes such as the gap genes and pair-rule genes, our preliminary analysis above is by no means intend to suggest significant biological findings. Instead, we use this as an example to demonstrate the potential utility of TESLA in offering a perspective of natural networks not receiving much attention before—the latent dynamic rewiring pattern of connectivities among nodes, which may shed light on the mechanisms of evolving behaviors of complex systems supported by the nodes.


1. Frank O, Strauss D (1986) Markov graphs. *J Am Stat Assoc* 81:832–842.
2. Wasserman S, Robins G (2005) An introduction to random graphs, dependence graphs, and p*. *Models and Methods in Social Network Analysis* (Cambridge Univ Press, Cambridge, UK).
3. Snijders T (2002) Markov chain Monte Carlo estimation of exponential random graph models. *J Soc Struct* 3(2).
4. Handcock M, Raftery A, Tantrum J (2007) Model-based clustering for social networks. *J R Stat Soc* 170(2):301–354.
5. Fienberg S, Airoldi E, Blei D, Xing E (2008) Mixed-membership stochastic blockmodels. *J Machine Learn Res* 9:1981–2014.
6. Snijders T (2001) The statistical evaluation of social network dynamics. *Sociological Methodology*, eds Sobel M, Becker M (Blackwell, Boston), pp 361–395, 2001.
7. Robins L, Pattison P (2001) Random graph models for temporal processes in social networks. *J Math Sociol* 25:5–41.
8. Hanneke S, Xing E (2006) Discrete temporal models of social networks. *Workshop on Statistical Network Analysis, the 23rd International Conference on Machine Learning (ICML-SNA)* (Association for Computing Machinery, New York).
9. Hoff P (2008) *Tech Rep no. 531, Department of Statistics* (Univ of Washington, Seattle, WA).
10. Friedman N (1998) The Bayesian structural em algorithm. *The 15th Conference on Uncertainty in AI* (Morgan Kaufmann, San Francisco).
11. Friedman N, Linial M, Nachman I, Pe'er D (2000) Using Bayesian networks to analyze expression data. *J Comput Biol* 77:601–620.
12. Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J R Stat Soc Ser B* 58(1):267–288.
13. Meinshausen N, Bühlmann P (2006) High-dimensional graphs and variable selection with the lasso. *Ann Stat* 34:1436.
14. Zhao P, Yu B (2006) On model selection consistency of lasso. *J Machine Learn Res* 7:2541–2563.
15. Ng A (2004) Feature selection, l1 vs. l2 regularization, and rotational invariance. *Proceedings of the 21st International Conference on Machine Learning* (American Society for Computing, New York).
16. Banerjee O, El Ghaoui L, d'Aspremont A (2008) Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J Machine Learn Res* 9:485–516.
17. Yuan M, Lin Y (2007) Model selection and estimation in the Gaussian graphical model. *Biometrika* 94(1):19–35.
18. Friedman J, Hastie T, Tibshirani R (2007) Sparse inverse covariance estimation with the graphical lasso. Tech rep (Stanford University, Stanford, CA).
19. Duchi J, Gould S, Koller D (2008) Projected subgradient methods for learning sparse Gaussians. *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence* (AUAI Press), pp 145–152.
20. Wainwright M, Ravikumar P, Lafferty J (2006) High dimensional graphical model selection using l1-regularized logistic regression. *Adv Neural Info Process Syst* 19:1465–1472.
21. Lee S, Ganapathi V, Koller D (2006) Efficient structure learning of Markov networks using l1-regularization. *Adv Neural Info Process Syst* 19:817–824.
22. Guo F, Hanneke S, Fu W, Xing E (2007) Recovering temporally rewiring networks: A model-based approach. *The 24th International Conference of Machine Learning, 2007* (Association for Computing Machinery, New York)
23. Zhou S, Lafferty J, Wasserman L (2008) Time varying undirected graphs. *Proceedings of the 21st Annual Conference on Learning Theory, 2008*, http://colt2008.cs.helsinki.fi/papers/COLT2008.pdf.
24. Dorman C, et al. (2007) Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. *Ecography* 30(5):609–628.
25. Ward M, Gleditsch K (2008) *Spatial Regression Models* (SAGE, Thousand Oaks, CA).
26. Anselin L, Florax R, ReyS (2004) *Advances in Spatial Econometrics: Methodology, Tools and Applications* (Springer, New York).
27. Martinussen T, Scheike T (2006) *Dynamic Regression Models for Survival Data* (Springer, New York).
28. Anselin L (1988) *Spatial Econometrics: Methods and Models* (Kluwer, Dordrecht, The Netherlands).
29. Haining R (2003) *Spatial Data Analysis: Theory and Practice* (Cambridge Univ Press, Cambridge, UK).
30. Tibshirani R, Saunders M, Rosset S, Zhu J (2005) Sparsity and smoothness via the fused lasso. *J R Stat Soc* 67:91–108.
31. Fruchterman T, Reingold E (1991) Graph drawing by force-directed placement. *Software Practice Experience* 21:1129–1164.
32. RyFiles W Organic layout. http://yfiles.net/products/yfiles/doc/developers-guide/organic_layouter.html.
33. Wills G (1997) Nicheworks—Interactive visualization of very large graphs. *Graph Drawing. Lecture Notes in Computer Science* (Springer, Berlin), pp 403–414.
34. Arbeitman M, et al. (2002) Gene expression during the life cycle of *Drosophila melanogaster*. *Science* 297:2270–2275.
35. Zhao W, Serpedin E, Dougherty E (2006) Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics* 22(17):2129–2135.

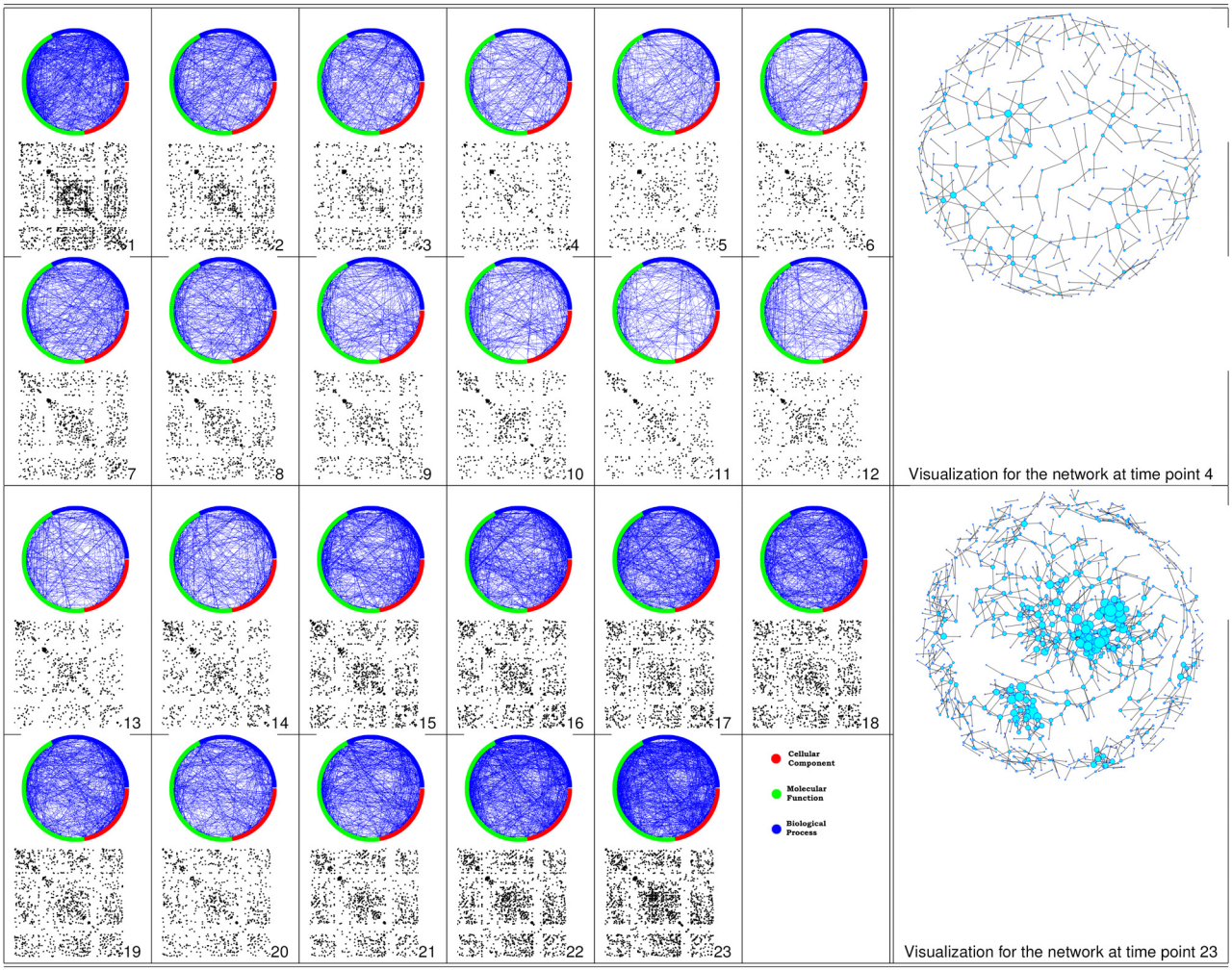


**Fig. S1.** The 23 snapshots of the dynamic gene interactions networks during the life cycle of the *D. melanogaster* are visualized by using a circular layout and a scatter plot of the network adjacency matrix. For both layouts, genes are ordered according to their top level functions (either related to cellular component, molecular function or biological process). Furthermore, 2 snapshots are further visualized using a spring layout algorithm and displayed in the last column of the figure.

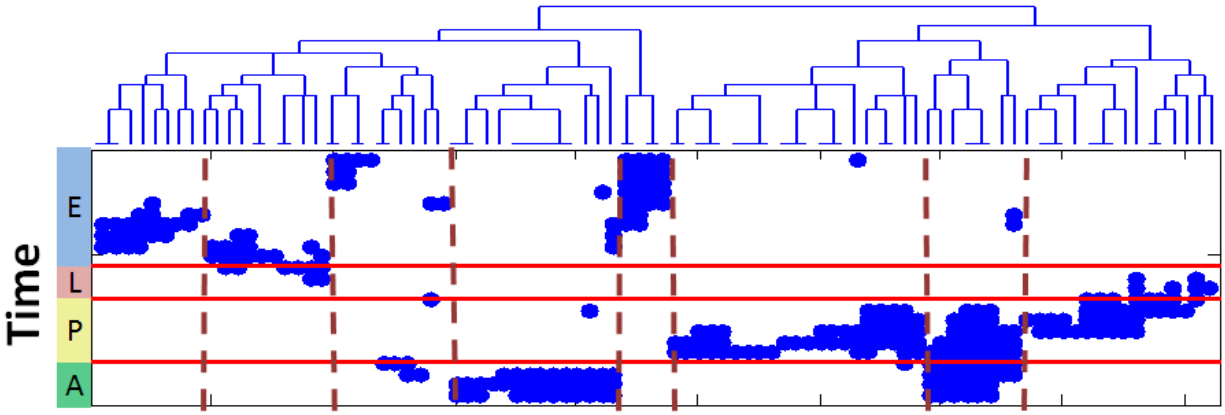


**Fig. S2.** A set of known gene interactions recovered by the inferred dynamic networks. The onset and duration of these interactions follow different temporal patterns. The activation of each gene interaction over time is represented as one column. Within each column, if a gene interaction is active a blue dot is drawn otherwise the space is left blank. Hierarchical clustering is performed on the gene interactions and clustering results are displayed on the top of the activation patterns which enables the visualization of block of gene interactions with similar activation patterns that deserves further biological investigations. The specific names of the genes involved in the interactions are not shown to avoid overcrowding the figure. E, L, P, and A stand for *D. melanogaster*'s developmental stages: embryonic, larval, pupal, and adulthood, respectively.

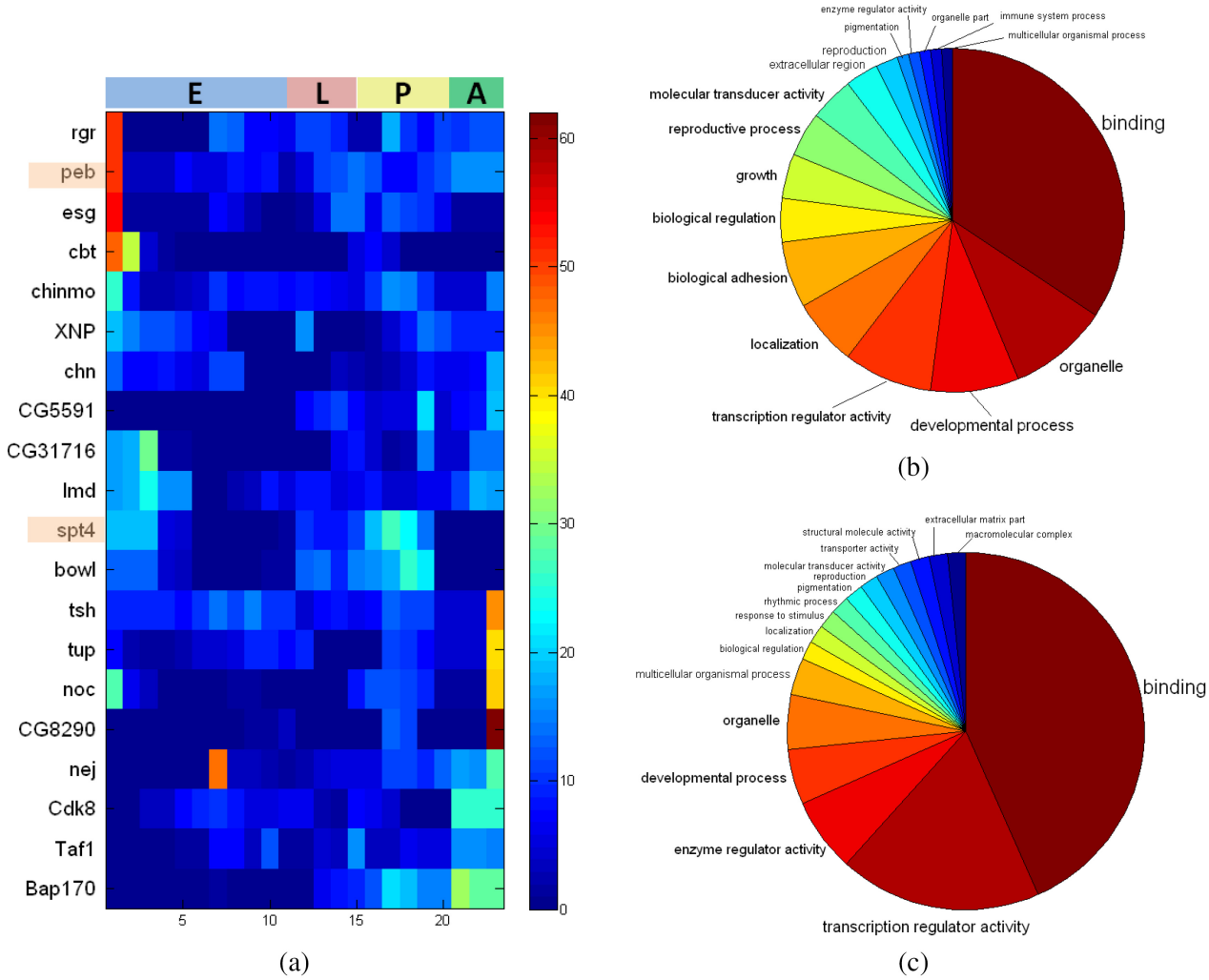


**Fig. S3.** (*A*) The top 20 TF hubs are tracked over time in terms of their degrees. Each row represents 1 hub, and each column represents a time point with the color code corresponds to the degree of the hub. (*B* and *C*) Examples of functional decomposition of the genes regulated by the transcriptional factor hub *peb* and *spt4*, respectively. E, L, P, and A stand for *D. melanogaster*'s developmental stages: embryonic, larval, pupal, and adulthood, respectively.

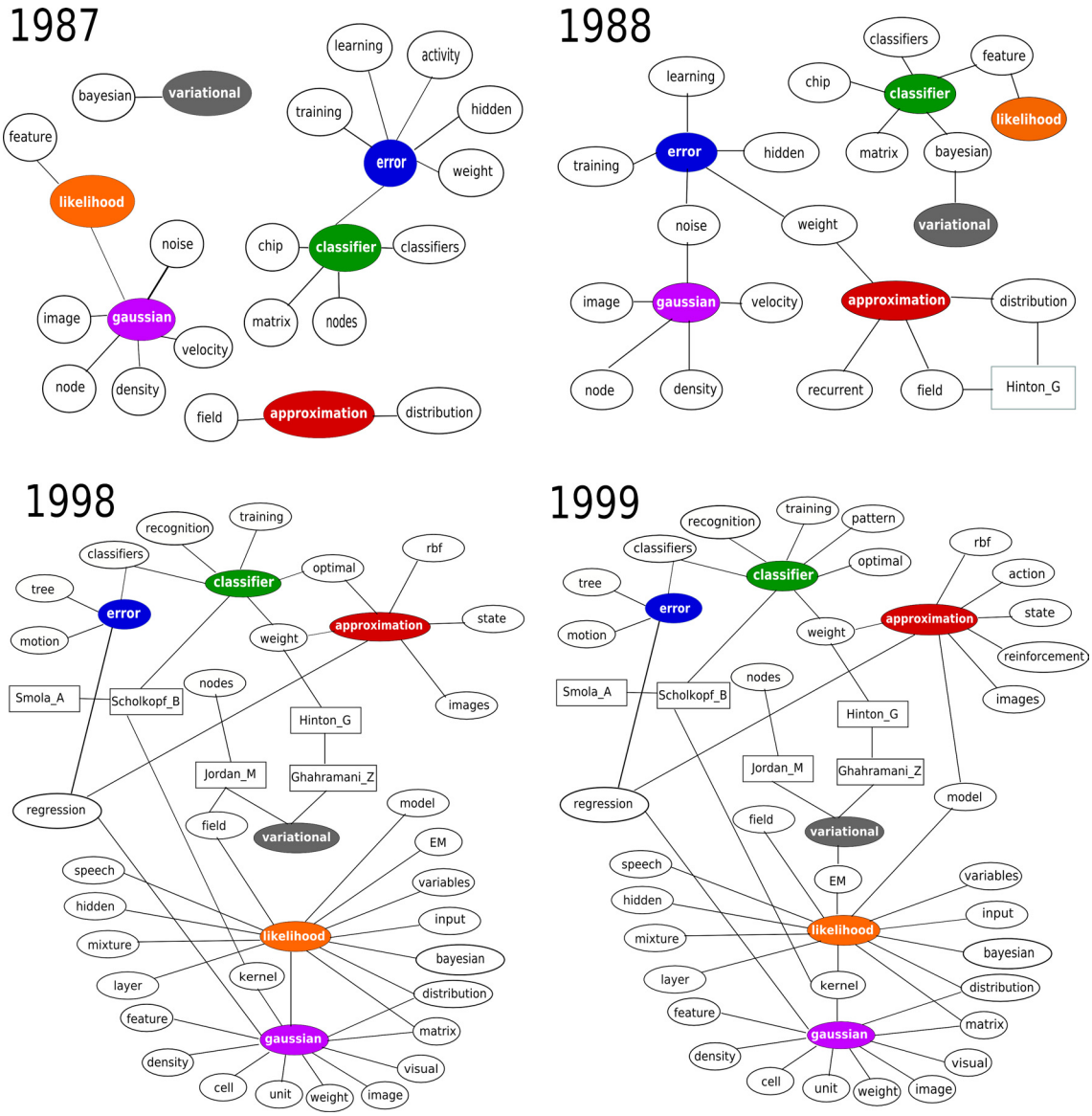


**Fig. S4.** Illustration of the NIPS academic social network from 1987 to 1999. Tracked words are highlighted, and authors are drawn in rectangles.