12-1-2021

Connecting the dots: The boons and banes of network modeling

Sharlee Climer
University of Missouri-St. Louis, climers@umsl.edu

Perspective

Connecting the dots: The boons and banes of network modeling

Sharlee Climer[1],*
[1]Department of Computer Science, University of Missouri – St. Louis, St. Louis, MO, USA
*Correspondence: climer@umsl.edu

https://doi.org/10.1016/j.patter.2021.100374

THE BIGGER PICTURE Deciphering high-dimensional patterns hidden in large datasets is a formidable undertaking due to the combinatorial explosion of the number of possible groups. A popular approach for tackling this challenge is network modeling, as it is a powerful and versatile technique that can be applied in a myriad of domains. A network is an abstraction of data in which each object is represented by a node, and an edge between a pair of nodes represents a relationship between the corresponding object pair. However, the full potential of this domain is not realized by current implementations due to several subtle, yet menacing, oversights. Here, we elucidate these flaws and provide commonsense solutions. Key issues include overextensions of the transitivity assumption, intolerance of subset heterogeneity, clustering biases, and mishandling of missing data. Solutions range from simple permutations and network scaffolding expansion to close examination and selections of pairwise relationship metrics and clustering algorithms. Application of these strategies reduces false-positive and false-negative signals and opens up opportunities to tease previously unidentified patterns concealed in the torrent of data produced across the sciences, industry, and government.

1 2 **3** 4 5 **Development/Pre-production:** Data science output has been rolled out/validated across multiple domains/problems

SUMMARY

Network modeling transforms data into a structure of nodes and edges such that edges represent relationships between pairs of objects, then extracts clusters of densely connected nodes in order to capture high-dimensional relationships hidden in the data. This efficient and flexible strategy holds potential for unveiling complex patterns concealed within massive datasets, but standard implementations overlook several key issues that can undermine research efforts. These issues range from data imputation and discretization to correlation metrics, clustering methods, and validation of results. Here, we enumerate these pitfalls and provide practical strategies for alleviating their negative effects. These guidelines increase prospects for future research endeavors as they reduce type I and type II (false-positive and false-negative) errors and are generally applicable for network modeling applications across diverse domains.

INTRODUCTION

Humans have aspired to infer knowledge by collecting and analyzing data for millennia. Such works include an ancient Sumer scientist c. 2000 BCE who created a data table, including row and column headers, and delineated information for a number of animals.[1] As the size of our global datasphere approaches 100 zettabytes, researchers in virtually every domain strive to harvest valuable information buried in a deep ocean of numerical and categorical data. Monumental data analysis advances have been achieved using machine learning, statistical, and operations research methods, yet accurately capturing complex patterns continues to challenge progress due to multiple factors. Key impediments include the sheer size of the search space, due to the combinatorial explosion of feasible patterns, and subtle assumptions underlying data analysis methods that may compromise outcomes.

Identification of high-dimensional patterns in data is inherently difficult due to the combinatorial explosion of the number of possible patterns (Table 1). Network modeling, also known as community detection, has emerged as a leading strategy in this conquest due to its scalability, flexibility, and ability to capture any order of relationship size. In this realm, a dataset is modeled as a network composed of nodes representing objects and edges representing relationships between the objects (Figure 1B).[2] In general, the edges can be directed to capture asymmetric relationships. In this article we are interested in undirected pairwise relationships and clustering methods based

Table 1. Example of combinatorial explosion

Size	1	2	3	4	k
No. of combinations	N	$\binom{n}{2} = \frac{n^2 - n}{2}$	$\binom{n}{3} = \frac{n^3 - 3n^2 + 2n}{6}$	$\binom{n}{4} = \frac{n!}{4!(n-4)!}$	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$
n = 1,000,000	1,000,000	499,999,500,000	1.7×10^{17}	4.2×10^{23}	$\frac{1,000,000!}{k!(1,000,000-k)!}$

Shown are the number of unique combinations for patterns comprising 1, 2, 3, 4, and k objects drawn from n objects, along with an example for a dataset with n = 1,000,000 objects.

on undirected edges, so we focus on symmetric relationships. Network analyses typically involve data pre-processing, computation of pairwise relationships, network construction, identification of clusters/communities within the network, and validation of results (Figure 1A).

Figure 1B illustrates Facebook and gene co-expression networks and Figure 1C describes characteristics for these networks, along with networks representing warehouse order picking and weather prediction. These examples illustrate the versatility of network modeling and provide illustrations for transferring real-world problems to a network structure. The Facebook network is a case in point of the strengths of network modeling. The input data are simply a list of an individual's Facebook friends and a list of pairs of these individuals that are Facebook friends with each other. Once these data are transformed into a network, clusters spontaneously arise. The numerous intra-cluster edges within a cluster indicate a high-ordered relationship and is the basis of the "guilt-by-association" postulation in this domain.[3] The transitivity assumption is at the heart of network modeling and provides the mechanism to infer high-ordered relationships from simple pairwise information.

Network modeling is capable of efficiently capturing high-ordered relationships, yet each step, from data pre-processing to validation of results, holds subtle impediments that arise due to intrinsic and extrinsic characteristics that may confound research progress. Here, we examine benefits and encumbrances of network modeling and demonstrate these characteristics in a popular application domain, gene co-expression analysis.[2,4–11] A brief description of this application follows.

Example network modeling problem: Gene co-expression analysis

A vigorous application domain for network modeling is gene co-expression analysis, which explores gene expression level data to identify patterns of genes that are synchronously expressing within one group of individuals more than another (Figure 1C).[12–15] Complex traits, such as disease states, arise due to aberrant biological pathways, many of which are not well understood. For example, the characteristic plaques that are hallmarks of late-onset Alzheimer disease are comprised of amyloid-β that is being overproduced, misfolded, and/or ineffectively cleared.[16–18] Identification of the deviant pathways underlying such processes facilitates understanding of the pathogenesis of diseases, revelations of unknown genetic functions, and recognition of potential drug targets.

Given expression levels of genes for a group of affected cases and a group of normal controls, the goal is to find patterns of co-expressed genes that appear significantly more often in one group than the other. Note that each individual gene may have similar mean levels in both groups. The challenge is twofold. First, synchronized patterns of multiple, perhaps hundreds, of genes that are co-expressing together within individuals must be extracted. Second, if an association with a trait is pursued, the percentages of individuals carrying the synchronized genetic pattern must be significantly different between the two groups. Exhaustive enumeration is not feasible due to the combinatorial explosion (Table 1). Gene co-expression analysis typically casts genes as nodes and places edges between pairs of genes that exhibit correlated expression across the individuals (Figure 1C). Clusters of co-expressed genes are identified and then evaluated for potential interactions and/or associations with the trait of interest.

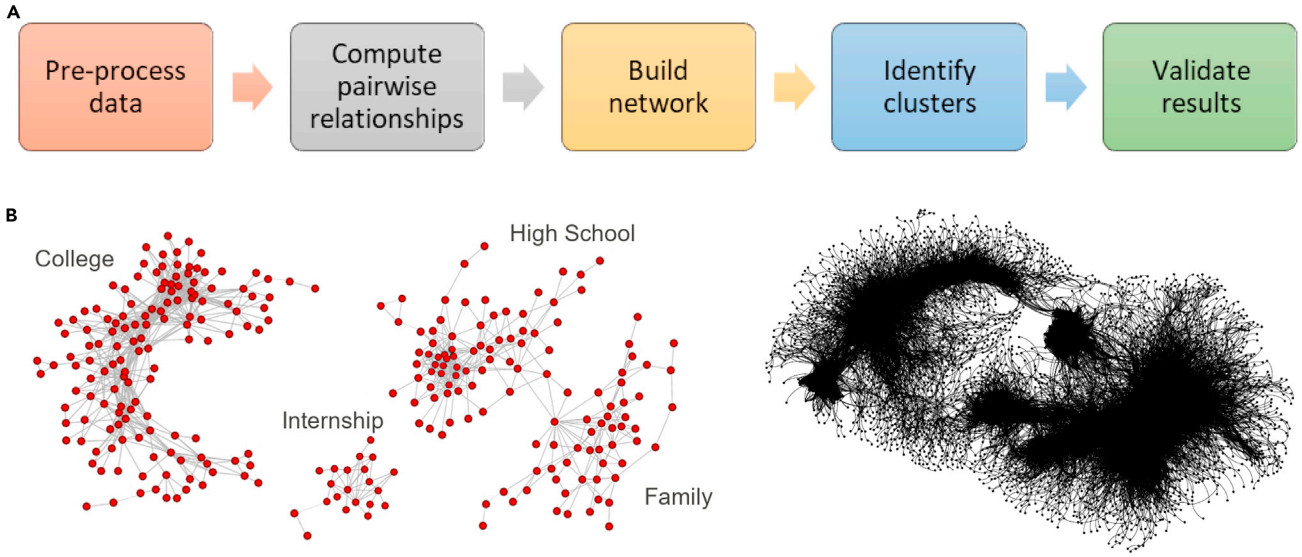
The organization of this article follows the steps usually taken for network modeling, with caveats for each step highlighted and potential remedies presented. We begin with data pre-processing, then discuss pairwise relationship computations, network construction, clustering, and validation. A brief discussion concludes the article.

DATA PRE-PROCESSING

Due to the massive size of most datasets of interest, it is not possible to manually inspect data before starting an analysis. Typos and improperly formatted data can silently sabotage a study, so it is important that software packages exit with meaningful error messages when encountered. Furthermore, outliers and missing data hold potential to quietly distort results. In general, data cleaning is challenging, and many steps are domain specific.[19] Here, we consider matters of general concern for network modeling: missing data and discretization, the latter of which palliates outliers.

Missing data

Missing data reduce power and potentially may lead to spurious correlations. Furthermore, some downstream analyses may require complete data. An approach that is receiving increasing popularity is data imputation, whereby the missing values are imputed based upon information drawn from the data. A wide range of methods have been developed, from simply replacing the missing values with the mean or median, to sophisticated methods designed to minimize the root-mean-squared error.[20–22] Local methods, such as K-nearest neighbors (KNNimpute)[23] and local least squares (LLSimpute),[24] identify similar objects via correlation metrics or Euclidean distance, to infer missing values. Global methods, such as Bayesian principal component analysis (BPCA),[25] disassemble the data and impute while rebuilding it. Classical methods, such as expectation maximization (EMimpute),[26] utilize incremental refinements while



C

	Facebook	Warehouse Order Picking	Weather	Gene Co-expression
Nodes	Facebook friends of a given individual	Merchandise in a online-store warehouse	Time points of weather attributes for a given region	Genes and their expression levels for cases with a given trait and normal controls
Edges	Binary; an edge is placed if the two friends are Facebook friends with each other	Discrete; number of times the two items were purchased together	Real-valued; correlation values based on weather attributes for the two time points	Real-valued; correlation values based on expression levels for the two genes over the group of individuals
Hub Node	Individual with exceptionally large number of Facebook friends	Popular merchandise item, when considering multi-item (and excluding single-item) orders	A time point with many attributes that are common for the given region	Regulatory gene, such as a gene that phosphorylates many other genes
Clusters	Social networks, such as a shared school, club, employer, or neighborhood	Merchandise items that commonly appear in orders together	Time points with similar attributes; a cluster containing many time points that preceded a given type of severe weather event would be of interest	Biological pathways; a cluster with a gene expression pattern that appears more often for cases than normal controls would be of interest
Applications	Targeted advertising - individuals in a shared cluster may exhibit similar purchasing preferences	Determination of shelf space allocation to increase order fulfillment packing efficiency; targeted advertising	Prediction of severe weather events	Hypothesis generation for advancing the understanding of biological processes underlying trait; drug targeting

Figure 1. Network modeling examples

(A) Typical steps in a network analysis.

(B) An example Facebook network (left) and gene co-expression network (right). For the Facebook network, each node represents a Facebook friend of a given individual, and an edge is placed between two nodes if the corresponding individuals are Facebook friends. For the gene co-expression network, nodes representing genes and edges are placed between two genes that exhibit correlated expression across a set of individuals.

(C) Four example network modeling applications. "Hub nodes" are nodes with exceptionally high degree.

iteratively maximizing likelihood. In general, these sophisticated methods outperform replacement with mean or median when assessed using the root-mean-squared error of the imputed values with the true values. However, this improvement may come with a cost for subsequent analyses which rely on correlations within the data. We next discuss three studies that investigate the impact of imputation error on downstream analyses.

Souto et al.[21] ran a series of trials to assess the impact of the four aforementioned imputation methods on downstream analyses. Using 12 cancer gene expression datasets, they imputed values with each method and then evaluated results for three network clustering algorithms. Interestingly, they observed that simply replacing values with the mean or median held similar performance as the four more elaborate techniques. They suggested that this observation may be due to the fact that clusters of co-expressed genes tend to be highly correlated and are likely to have some genes with no missing data, hence high accuracy of imputed values is not critical in downstream analyses.

We propose an alternative viewpoint. A key stumbling block for data imputation prior to network modeling is that error in the imputations is *not* random for approaches that use correlations, such as KNNimpute, LLSimpute, BPCA, and EMimpute. When relationships within the data are used, exceptions to the trends are erroneously replaced with values that match the observed patterns. These biased errors can falsely boost pairwise relationships that are used to create edges for the network. In short, while the overall root-mean-squared error may be lower when one of these methods is utilized as opposed to simply using the mean or median, the inaccuracies that do arise tend to increase downstream correlation values and false-positive errors.

A second study focused on the effects of imputation on an analysis of questionnaire data based on stress and health for older adults.[22] This 20-page survey instrument included questions for computing scores for symptoms of depression, anxiety, and self-assessed health. A set of 96 cases with no missing data had the computed score for symptoms of depression removed from the data, along with 19.5% of data points used to compute this score. The missing data were imputed using simple regression (SR), regression with added error term (RET), and expectation maximization (EM), and the imputed score for depression symptoms obtained. The correlation between the imputed depression score and three of the variables included in the score calculations—sex, age, and self-assessed health—were computed for the original data and for the data following the three imputation methods. The authors also computed correlations between the depression score and two scores not included in the imputations: anxiety and functional health. While these two scores had strong correlations with the depression scores ($p \leq 0.001$ for anxiety and $p \leq 0.01$ for functional health) in the original data, the imputed depression scores exhibited dramatic differences. EM showed high significance ($p \leq 0.01$) in the opposite direction for anxiety and SR showed high significance ($p \leq 0.05$) in the opposite direction for functional health. The three variables with the imputed values sex, age, and self-assessed health did not exhibit this type of reversed correlation. The correlation between depression scores and sex for EM imputation was similar to the original score, while SR and RET failed to capture any significant correlation. Both age and self-assessed health demonstrated strong inflation of the correlation. Age was not correlated with depression for the original data and was significantly correlated for RET ($p \leq 0.05$) and EM ($p \leq 0.01$) in the imputed data. Self-assessed health was significantly anti-correlated ($p \leq 0.05$) with depression in the original data, uncorrelated for RET, and jumped to very strong anti-correlation for both SR and EM ($p \leq 0.001$). In short, the variables with imputed values tended to boost correlation values, while those without imputations exhibited unpredictable correlations with the imputed depression score.

The third study examined the effects of imputation on mass spectrometry data taken across various tissues.[27] For each tissue, data values were imputed using seven different imputation methods (half minimum substitution, mean substitution, *k*-nearest neighbors, local least-squares regression, BPCA, singular value decomposition, and random forest). Following imputations at levels of missingness ranging from 10% to 50%, correlations between the matrices were computed and MANOVA trials were run. The authors observed two primary outcomes. First, the magnitude of the pairwise inter-matrix correlations declined and in some cases reversed in direction. Presumably this is due to erroneous inflation of the correlation patterns within each of the matrices induced by the imputations. Second, the number of false-positive errors in the MANOVA tests increased in accordance with the level of missingness for all seven imputation methods.

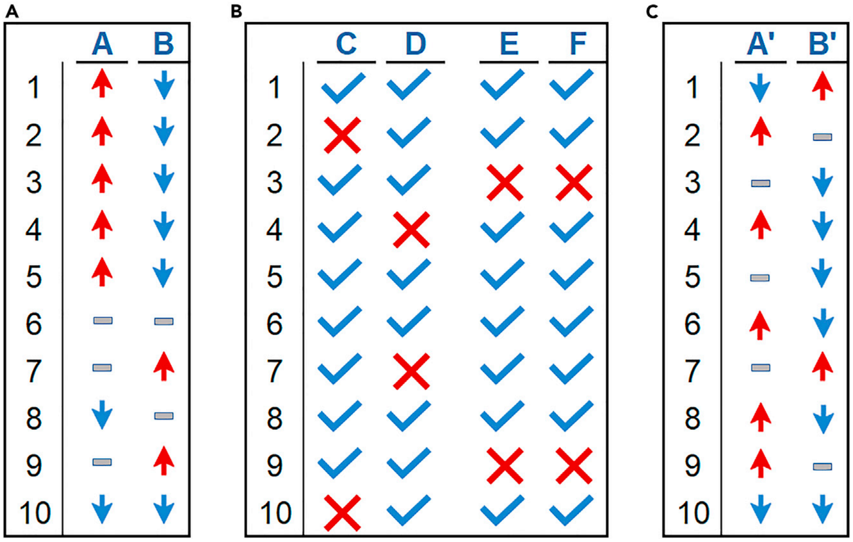
In summary, data imputation methods that draw on patterns that exist in the known data points tend to reinforce these relationships, thereby inflating correlation structures, and hold potential to produce false-positive edges in network models. Data imputation may be more useful in approaches that are not based on network construction. For example, genome-wide association studies, whereby each genetic marker is directly analyzed for association with a trait and the relationships between the markers are not computed, may be more resilient to bias in imputation error.

In lieu of imputation, a common approach is to remove any rows or columns of the data table with excessive missing data values.[28] The downside is that a lot of known data points are lost in this process. It should be noted that starting with a relaxed threshold for missingness and iteratively removing rows and columns in an alternating fashion while gradually tightening the threshold often leads to higher data retention than applying the target threshold to all rows and columns simultaneously. We use an Alzheimer disease gene expression dataset[29] generated by Amanda Myers' lab to demonstrate. These data include expression levels for 8,650 genes drawn from 363 individuals' postmortem brain. Directly cleaning to a maximum of 5% missing values for all individuals and for all genes eliminates 1,243 genes and 46 individuals. On the other hand, using the iterative procedure while striving to retain individuals eliminates 1,219 genes and 6 individuals. We offer open-source software for facilitating this iterative process at www.blocbuster.org.

Another consideration is the relative distribution of the missing values between the two objects being measured for a relationship, as will be presented in the next section.

Discretization

Discretization of data values, whereby real values are binned into a set of discrete values such as low, average, and high, is performed in many analyses. Such techniques can facilitate computations, tolerate differences in scales across objects, and eliminate outlier concerns.[30,31] However, the choice of discretization thresholds may have dramatic effects on results.[30] When re-running entire analyses using different discretization thresholds is impractical, it is advisable to check the sensitivity of the results using different thresholds. While network analysis methods may benefit from the use of discretized data, it may be practical to assess the results found using the original continuous-valued data. When this type of validation is conducted, outliers should be carefully treated using an appropriate method that accounts for specific intricacies arising in the given research area. For example, in the gene expression domain, rare genetic variants can yield outlier gene expression values that are indeed biologically relevant.[32]



	A	B
1	↑	↓
2	↑	↓
3	↑	↓
4	↑	↓
5	↑	↓
6	–	–
7	–	↑
8	↓	–
9	–	↑
10	↓	↓

	C	D	E	F
1	✓	✓	✓	✓
2	✗	✓	✓	✓
3	✓	✓	✗	✗
4	✓	✗	✓	✓
5	✓	✓	✓	✓
6	✓	✓	✓	✓
7	✓	✗	✓	✓
8	✓	✓	✓	✓
9	✓	✓	✗	✗
10	✗	✓	✓	✓

	A'	B'
1	↓	↑
2	↑	–
3	–	↓
4	↑	↓
5	–	↓
6	↑	↓
7	–	↑
8	↑	↓
9	↑	–
10	↓	↓

Figure 2. Subset heterogeneity, effective sample size, and permutation examples

Examples for pairs of objects, each with ten attribute values. Red upward arrow, dash, and blue downward arrow indicate high, neutral, and low data values, respectively. An "×" indicates missing data value.

(A) The first five attribute values are perfectly correlated for objects A and B, while the other five are not correlated at all. Such a situation may be expected in the presence of subset heterogeneity. The absolute value of Pearson's correlation coefficient is only 0.44 due to the uncorrelated values. Duo returns a high score of 0.80 for the high/low relationship and low scores for high/high, low/high, and low/low relationships.

(B) Objects C, D, E, and F each have 20% missing data. When computing a pairwise correlation measure for objects C and D, 40% of the value pairs contain missing values and do not contribute to the score. On the other hand, only 20% of the value pairs contain missing values for objects E and F.

(C) A′ and B′ represent random permutations of objects A and B, respectively. Each object retains the same values while the inherent correlation between A and B is broken up.

PAIRWISE RELATIONSHIP CALCULATIONS

After pre-processing data, pairwise relationships are computed to generate edges in the network. The number of computations to assess all pairs is equal to $(n^2 - n)/2$, where n is the number of objects. Given an efficient algorithm and adequate resources, this number is feasible for many datasets of interest. When the computation time is too burdensome, these independent pairwise computations can be run in parallel across many processors, and cloud services are readily available for such tasks.

As illustrated in Figure 1C, edges may be binary or carry a discrete or real-valued weight. The Facebook network example includes binary edges, where an edge exists if the individuals are Facebook friends and does not exist otherwise. Most network models of interest require a more complex evaluation of pairwise relationships. Similarity or correlation measures computed across arrays of values representing each object, such as Euclidean distance or Pearson's correlation coefficient (PCC),[33] are commonly utilized, but some applications may require a domain-specific relationship computation. We discuss four challenges regarding this step: subset heterogeneity, sample size, spurious correlations, and edge retention.

Subset heterogeneity

Many network modeling domains exhibit subset heterogeneity, and such heterogeneity should be addressed by the correlation metric utilized. Examples of subset heterogeneity include different weather patterns preceding a common severe weather event and subtypes of diseases, such as breast cancer, in which different biological pathways are manifesting a shared cancer phenotype. Not only is it valuable to tease out these different subgroups to increase weather prediction accuracy and facilitate precision medicine, failure to account for this heterogeneity can yield false-negative correlations, as shown in Figure 2A. Prominent correlation measures, such as PCC and Euclidean distance, return a single scalar value that must account for the correlation over all of the data points in the arrays. This is problematic as when heterogeneity exists, one subgroup may exhibit high correlation, but there is no reason to expect other subgroups to hold any correlation, and this lack of correlation tends to weaken the correlation score. The only correlation measures that we are aware of that account for subset heterogeneity are Hamming distance[34] and its variants, and the two vector-based correlation measures that we have introduced: custom correlation coefficient[35,36] for single-nucleotide polymorphism data and Duo[11] for general real-valued data.

Sample size

Inadequate sample size increases the likelihood of observing spurious correlations and false-positive signals. Spurious correlations generally fall into two categories: those that arise from an indirect relationship and those that arise by mere chance. The first of these types can be expected in network analysis. For example, two genes may be exhibiting high expression together due to an underlying biological condition. Here, we consider spurious correlations that arise by mere chance.

In general, a sample size that will adequately diminish spurious correlations can be difficult to correctly ascertain, as it is highly dependent upon the properties of the given dataset and the correlation metric employed. Moreover, for a given sample size and correlation metric, the effective sample size can be reduced due to missing data, with the reduction being dependent upon the relative locations of the missing data values. For example, consider PCC. This popular correlation measure is based on the covariance of the two arrays divided by the product of the standard deviations for the arrays. It should be noted that the percentage of missing values for each object is only a lower bound on the level of missing values used in correlation computations, as they range from the maximum percentage of the two objects to the sum of the percentages for the two objects, as shown in Figure 2B. In essence, the effective sample size can vary between each pair of objects. It is desirable for software to report warnings when the effective sample size drops below a given threshold, yet such features are rare.

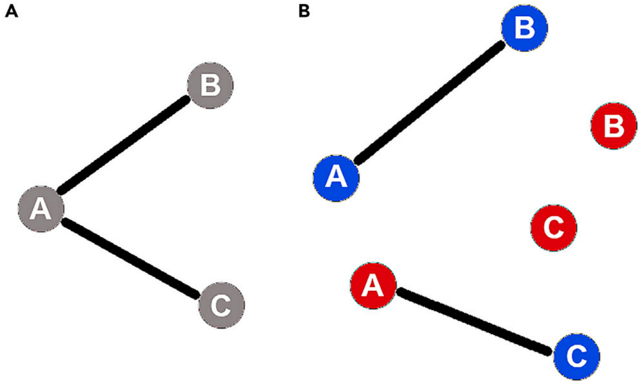


Figure 3. Duality node
Assume that low values of object A are correlated with low values of object B, high values of object A are correlated with low values of object C, and no other correlations exist for objects A, B, and C.
(A) In a standard network for which each object is represented by a single node, the transitivity assumption would falsely suggest that B and C are correlated.
(B) In an expanded network for which each object is represented by two nodes, one for high values and one for low values (red and blue, respectively), B and C are not joined by an intermediate node.

Spurious correlations

In addition to inadequate effective sample size, spurious correlations can arise due to characteristics of the data and the algorithm employed. An agile approach to dynamically test for these errors is to run permutation trials for each pair of objects, thereby testing the null hypothesis for the given pair. For each correlation measurement above a given threshold, the corresponding pair of objects has their values permuted as shown in Figure 2C for an adequately large number of trials (e.g., 1,000). These permutations break up inherent correlations that might exist while retaining sample size and other statistical properties of each array, such as median and variance, as they are composed of exactly the same values but in different relative ordering. Correlation is measured over the permuted arrays and sorted to yield a p value for the degree of correlation for the array pair.

Edge retention

The number of possible edges in a network with n nodes is $(n^2 - n)/2$. As it is not practical to hold all edges of a complete network in main memory for all but small n, a large proportion of edges is not retained. Assuming permutation trials are run, a minimum criterion for edge retention might be to require a p value of less than 0.05.

NETWORK CONSTRUCTION

The construction of a network once the edges have been identified is relatively straightforward. However, there is an insidious fundamental mistake that is practiced nearly universally, as described in this section. Another challenge is assessing the structure of the network, which is also addressed herein.

Duality nodes

Networks are normally constructed by assigning a node to represent each object and placing edges between pairs of nodes that are correlated. This practice leads to false-positive signals due to the transitivity assumption upon which network modeling is based. For a given object, correlations with other objects can arise due to high or low values in the object's array of data. For instance, high and low values of temperature, atmospheric pressure, wind, precipitation, cloudiness, and/or humidity are each associated with different weather events. Note that high values for one object and low values of another may be involved in important anti-correlations. Typical scalar correlation metrics indicate the degree of correlation/anti-correlation but do not indicate whether high or low values are contributing to the relationship, creating an environment for the generation of what we refer to as duality nodes (Figure 3).[11] Duality nodes lead to the merging of unassociated clusters. Moreover, these clusters may be the opposite of each other. For example, if high expression of gene A is correlated with a cluster of genes that lie in a biological pathway leading to disease progression and low expression of A is correlated with a healthy biological pathway, the genes for both of these opposing pathways will be connected via A. Consider the β-site amyloid precursor protein (APP)-cleaving enzyme 1 (BACE1). BACE1 competes with α-secretase ADAM10 for cleaving APP.[37] While ADAM10 cleavage has not been associated with deleterious effects, BACE1 cleavage yields β-amyloid peptides, which aggregate to form the amyloid plaques that are characteristic of Alzheimer disease.[38] High expression of BACE1 has been observed in peripheral blood of Alzheimer disease cases when compared with normal controls.[39] Consequently, a network in which each gene is represented by a single node will tend to connect the pathological pathway yielding production of excess β-amyloid peptides with analytes in healthy pathways that include low BACE1 levels. We have addressed this issue by expanding network scaffolding to include two nodes per object, representing high and low values, respectively.[11] As illustrated in Figure 3B, this expansion separates the clusters and justifies the use of transitivity.

Allocating two nodes for each object doubles the number of nodes needed, but the number of edges is not increased. Indeed the resulting network is somewhat sparsified, and large connected components may be separated into smaller connected components. Network clustering is typically the most computationally demanding step during a network analysis, and each separate connected component can be clustered independently without any loss of accuracy. Identification of the connected components can be quickly computed using a modified breadth-first search (BFS) that runs in $O(n + e)$ time, where n is the number of nodes and e is the number of edges.[40] (We provide open-source code for this purpose at www.blocbuster.org.) In summary, while network expansion doubles the number of nodes, it eliminates false-positive signals due to duality nodes while retaining the same number of edges and may reduce the computational demands for downstream clustering analyses.

Network structure assessment

Large-scale networks are difficult to visualize due to their complexity and high dimensionality. Many visualization tools exist, such as Gephi[41] and Cytoscape,[42] along with Python, R, and MATLAB tools, but they tend to be computationally demanding and typically are unable to render large networks of interest. Moreover, these programs attempt to flatten a high-dimensional network into two-dimensional (2D) space, and this dimension squashing can obscure interesting characteristics. There are many different algorithms for laying out a network in two dimensions, such as Force Atlas,[41] Fruchterman-Reingold,[43] and Yifan Hu,[44] and these methods generally yield vastly different visualizations that do not even appear to represent a common network. Consequently, it is advisable to view multiple layouts and to also consider other resources, as follows.

To gain insights into large-scale network structure, one can identify properties such as edge density, node degree distributions, reciprocity, bridge counts, and centrality.[45] In our genetics research, we have observed many networks that contain large numbers of singleton nodes without any edges connecting them to any other nodes, and completely disconnected components, in which no edges connect the components to each other. Knowledge of such structures can simplify downstream analysis by removing singletons and assessing each component separately, thereby reducing computational burden. As previously mentioned, a BFS of the network can be adapted to explore the network, thereby providing the numbers of nodes and edges for each disconnected component and a count of singleton nodes. Networks with disconnected subcomponents can be separated into smaller networks, each of which may be manageable for visualization tools.

CLUSTERING

Typical clustering algorithms are not easily parallelized, and the computational bottleneck in a study may arise in this step. For this reason, it is common for researchers to prune the objects that appear the least promising. However, it is difficult to know a priori which objects to choose, as excluded objects may play roles in valuable synergistic interactions. An alternative approach is to increase the edge retention stringency to decrease the number of edges until the network breaks into disconnected components. After the components are identified, the discarded edges within each component can be replaced. Consequently, each component will require less computation time than the original network and can be run in parallel on different processors.

Identifying an unknown number of clusters, also referred to as communities or modules, each with an unknown number of tightly connected nodes, can be a daunting task. A plethora of algorithms have arisen using diverse computational tools. Once an algorithm is selected, there are typically multiple adjustable parameters yielding a great variety of outputs. Taken together, there is a vast number of clustering results possible, which presses the question: which is correct for your network? Many researchers rely on precedence and simply use clustering algorithms and parameter settings that have been published in their domains previously. However, those previous selections may have been somewhat arbitrary and/or differences in network structures may invalidate this reuse.

For algorithms that are not based on a specific objective, underlying assumptions and objectives are often difficult to assess, despite their importance for method selection. For example, many popular clustering methods, including *k*-means[46] and hierarchical clustering,[47] assume clusters have hyperspherical shapes and tend to minimize the overall diameters of the clusters. Many practical applications may yield elongated or complex structures that are likely to be cut apart by the sphericity assumption. Also, differences in densities of clusters within a single network can impede some algorithms, such as DBSCAN.[48,49]

Some clustering methods are based on clearly stated objectives. For example, a large group of clustering algorithms aim to maximize the modularity function that was proposed by Newman and Girvan in 2004.[50] Modularity measures the numbers of edges within assigned clusters minus the numbers expected if the edges are placed randomly, while node degrees remain constant. This objective does not enforce sphericity and gained rapid popularity. Optimally maximizing the modularity objective function is NP-hard[51] so many approximation implementations have arisen, including greedy methods,[50] divisive optimization,[52] simulated annealing,[53] hierarchical clustering,[54] and spectral partitioning.[55] Sixteen different modularity implementations have been compared by Danon et al.[56] Fortunato and Barthélemy observed a resolution limit for modularity wherein distinct clusters will be merged together when the network size is adequately large.[57] We have observed that modularity is strongly biased against singletons, regardless of network size, and will sometimes split a dense cluster in two to avoid creating a singleton cluster. Consequently, modularity-based methods may be problematic for networks in which singletons are expected and for very large networks.

In many research endeavors it is not clear what clustering objective is suitable, and it is tempting to apply many different clustering methods. However, multiple testing corrections should be applied, making this expedition prohibitive. Lea and Climer developed a solution to this dilemma by applying many different clustering techniques and sorting the clusters by desirable properties to select the most promising for validation testing, thereby managing multiple testing corrections.[58] Another resource is VICTOR (http://bib.fleming.gr:3838/VICTOR/). This website provides visualizations of various clustering algorithms to aid in cluster selection.[59]

VALIDATION

Using an adequate number of permutation trials for pruning false-positive correlations, representing each object by two nodes to eliminate duality nodes, and utilizing an appropriate correlation metric and clustering technique will increase the likelihood of correct results. However, noise in the data and overfitting can sabotage outcomes, and it is imperative that results are validated.

Depending on research design, validation via data generated by a different study may be problematic due to differences in data collection. In the realm of gene expression data, differences in platforms used to measure gene expression alone can be drastic enough to undermine efforts, as different variants of each gene may be captured. Furthermore, sample preparation, technician experience, and equipment settings can yield inconsistencies between studies.[60] Alternatively, many publications report gene enrichment p values as validation of the results. Various reference databases, such as DAVID[61] and Metascape,[62] provide software to estimate the probability of seeing a group of biologically related genes appearing in a given module of genes.[63] These results are dependent upon the clustering algorithm utilized by the software and the number and sizes of clusters.[10] Furthermore, the analysis is based entirely on known biological relationships and, consequently, novel discoveries will not fare well in these evaluations.

In general, it is ideal to split the data samples into discovery and validation sets, use the discovery data to generate the network and clusters, and test these clusters in the held-out validation data. For example, 70% of the samples can be used to build the network and discover patterns associated with the trait of interest, then each of these patterns can be tested for associations on the held-out samples, with multiple testing corrections applied. Unfortunately, this approach can diminish the power needed to identify true correlations and clusters in the discovery dataset while ensuring that the validation dataset is adequately large to be representative of the true patterns in the data. However, many data collection methods are becoming increasingly more affordable, and datasets are growing to suitable sizes in many domains.

DISCUSSION

The pearls and pitfalls of network modeling are numerous. The beauty of the approach is that arbitrarily high-dimensional patterns can be identified based upon simple pairwise relationships. Given an efficient implementation and adequate computational resources, it is feasible to build and analyze networks for most datasets of interest. Another advantage is that components of a network can be visualized using 2D and 3D plotting software. These visualizations capture complex interactions and may reveal interesting characteristics worthy of further exploration, such as hub nodes that are connected to large numbers of other nodes and/or dense subclusters that are loosely connected.

As detailed herein, there are numerous caveats that are commonly overlooked in network modeling. First, imputation of missing data can lead to false-positive signals for downstream correlation measurements. An alternative strategy is to iteratively remove objects and attributes with excessive numbers of missing values while gradually tightening the threshold until a desired threshold is reached.

Second, discretization of data values, when utilized, needs to be evaluated for robustness of the discretization thresholds utilized.

Third, the pairwise relationship metric must align with the specific properties of the domain. In particular, a common error is to apply a general-purpose correlation measure when subset heterogeneity exists, thereby leading to false-negative signals.

Fourth, the "sample size" for a study is not necessarily equal to the effective sample size. For each pairwise relationship computation, the effective sample size is dependent upon the amount and the relative positioning of the missing data for the pair.

Fifth, spurious correlations may arise. A straightforward strategy for assessing significance to use permutation trials and then base edge retention on the p values derived from an ample number of such trials.

Sixth, duality nodes are pervasive and dicey actors hidden in the network modeling realm. It is natural to represent each object as a node, yet, in hindsight, it is clear that "high" and "low" values of an object should not be compressed into a single node, as it invalidates the transitivity assumption upon which network modeling is based.

Seventh, although plotting network subcomponents can be insightful, visualization of high-dimensional networks in 2D space is somewhat arbitrary. Evaluating network properties can yield meaningful information while providing statistical characteristics to inform the next step: clustering.

Eighth, properly clustering the network can be a daunting task. It is possible to ameliorate computational demands using divide-and-conquer strategies. However, selection of a valid clustering algorithm from the profusion of offerings, along with appropriate parameter settings, is challenging and requires careful considerations of the particular structure of the given network.

Finally, despite best practices in network analysis, false-positive signals may arise due to noise in the data and overfitting. Stringent unbiased validation is indispensable and can be achieved using independent data.

While these many challenges can be assuaged using the prescribed techniques, one pressing issue is that regardless of the perfection of the analysis, network modeling is an approximation method. Even if a dataset is analyzed a very large number of times using many different choices, there is never any guarantee that all useful patterns are revealed, and the most beneficial signals may remain hidden within the sea of values. The number of possible patterns grows exponentially with the pattern size (e.g., patterns of sizes 2, 3, and k have the order of n^2, n^3, and n^k possible patterns for n nodes, as shown in Table 1). Consequently, ensuring optimality is expected to be intractable for problem sizes of interest given currently available methods. Fortunately, when properly applied, the guilt-by-association basis of network modeling provides a scalable and flexible vehicle for releasing insightful high-dimensional relationships from otherwise incomprehensible datasets.

ACKNOWLEDGMENTS

This work was supported by National Institute on Aging grants 1RF1AG053303-01 and 3RF1AG053303-01S2.

REFERENCES

1. Kidwell, P.A. (2004). A history of mathematical tables: from Sumer to spreadsheets. Technol. Cult. *45*, 662–664.
2. Jones, P., Weighill, D., Shah, M., Climer, S., Schmutz, J., Sreedasyam, A., Tuskan, G., and Jacobson, D. (2019). Network modeling of complex data sets. In Metabolic Pathway Engineering: Methods and Protocols, M. Himmel and Y. Bomble, eds. (Springer), pp. 197–215.
3. Quackenbush, J. (2003). Genomics. microarrays—guilt by association. Science *302*, 240–241.
4. Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. *95*, 14863–14868.
5. Lee, H.K., Hsu, A.K., Sajdak, J., Qin, J., and Pavlidis, P. (2004). Coexpression analysis of human genes across many microarray data sets. Genome Res. *14*, 1085–1094.
6. Langfelder, P., and Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. BMC Bioinform. *9*, 559.
7. Choi, J.K., Yu, U., Yoo, O.J., and Kim, S. (2005). Differential coexpression analysis using microarray data and its application to human cancer. Bioinformatics *21*, 4348–4355.

8. Oldham, M.C., Konopka, G., Iwamoto, K., Langfelder, P., Kato, T., Horvath, S., and Geschwind, D.H. (2008). Functional organization of the transcriptome in human brain. Nat. Neurosci. *11*, 1271–1282.

9. Stuart, J.M., Segal, E., Koller, D., and Kim, S.K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. Science *302*, 249–255.

10. Ruan, J., Dean, A.K., and Zhang, W. (2010). A general co-expression network-based approach to gene expression analysis: comparison and applications. BMC Syst. Biol. *4*, 8.

11. Climer, S., Templeton, A.R., Garvin, M., Jacobson, D., Lane, M., Hulver, S., Scheid, B., Chen, Z., Cruchaga, C., and Zhang, W. (2020). Synchronized genetic activities in Alzheimer's brains revealed by heterogeneity-capturing network analysis. bioRxiv. https://doi.org/10.1101/2020.01.28.923730.

12. van Dam, S., Võsa, U., van der Graaf, A., Franke, L., and de Magalhães, J.P. (2018). Gene co-expression analysis for functional classification and gene-disease predictions. Brief. Bioinform. *19*, 575–592.

13. Horvath, S., and Dong, J. (2008). Geometric interpretation of gene coexpression network analysis. Plos Comput. Biol. *4*, e1000117.

14. Fuller, T., Langfelder, P., Presson, A., and Horvath, S. (2011). Review of weighted gene coexpression network analysis. Handb. Stat. Bioinforma. 369–388.

15. Kopp, N., Climer, S., and Dougherty, J.D. (2015). Moving from capstones toward cornerstones: successes and challenges in applying systems biology to identify mechanisms of autism spectrum disorders. Front. Genet. *6*, 301.

16. Mawuenyega, K.G., Sigurdson, W., Ovod, V., Munsell, L., Kasten, T., Morris, J.C., Yarasheski, K.E., and Bateman, R.J. (2010). Decreased clearance of CNS β-amyloid in Alzheimer's disease. Science *330*, 1774.

17. Selkoe, D.J., and Hardy, J. (2016). The amyloid hypothesis of Alzheimer's disease at 25 years. EMBO Mol. Med. *8*, 595–608.

18. Haass, C., and Selkoe, D.J. (2007). Soluble protein oligomers in neurodegeneration: lessons from the Alzheimer's amyloid β-peptide. Nat. Rev. Mol. Cell Biol. *8*, 101–112.

19. Chu, X., Ilyas, I.F., Krishnan, S., and Wang, J. (2016). Data cleaning: overview and emerging challenges. In Proceedings of the ACM SIGMOD International Conference on Management of Data (Association for Computing Machinery), pp. 2201–2206.

20. Lin, W.C., and Tsai, C.F. (2019). Missing value imputation: a review and analysis of the literature (2006-2017). Artif. Intell. Rev. *532*, 1487–1509.

21. Souto, M.C.P.D., Jaskowiak, P.A., and Costa, I.G. (2015). Impact of missing data imputation methods on gene expression clustering and classification. BMC Bioinform. *16*, 64.

22. Musil, C.M., Warner, C.B., Yobas, P.K., and Jones, S.L. (2016). A comparison of imputation techniques for handling missing data. West. J. Nurs. Res. *24*, 815–829. https://doi.org/10.1177/019394502762477004.

23. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R.B. (2001). Missing value estimation methods for DNA microarrays. Bioinformatics *17*, 520–525.

24. Kim, H., Golub, G.H., and Park, H. (2005). Missing value estimation for DNA microarray gene expression data: local least squares imputation. Bioinformatics *21*, 187–198.

25. Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., and Ishii, S. (2003). A Bayesian missing value estimation method for gene expression profile data. Bioinformatics *19*, 2088–2096.

26. Bø, T.H., Dysvik, B., and Jonassen, I. (2004). LSimpute: accurate estimation of missing values in microarray data with least squares methods. Nucl. Acids Res. *32*, e34.

27. Taylor, S.L., Ruhaak, L.R., Kelly, K., Weiss, R.H., and Kim, K. (2017). Effects of imputation on correlation: implications for analysis of mass spectrometry data from multiple biological matrices. Brief. Bioinform. *18*, 312–320.

28. Raymond, M.R., and Roberts, D.M. (1987). A comparison of methods for treating incomplete data in selection research. Educ. Psychol. Meas. *47*, 13–26.

29. Webster, J.A., Gibbs, J.R., Clarke, J., Ray, M., Zhang, W., Holmans, P., Rohrer, K., Zhao, A., Marlowe, L., Kaleem, M., et al. (2009). Genetic control of human brain transcript expression in Alzheimer disease. Am. J. Hum. Genet. *84*, 445–458.

30. Yang, Y., Webb, G.I., and Wu, X. (2009). Discretization methods. In Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, eds. (Springer), pp. 101–116.

31. Liu, H., Hussain, F., Tan, C.L., and Dash, M. (2002). Discretization: an enabling technique. Data Min. Knowl. Discov. *6*, 393–423.

32. Li, X., Kim, Y., Tsang, E.K., Davis, J.R., Damani, F.N., Chiang, C., Hess, G.T., Zappala, Z., Strober, B.J., Scott, A.J., et al. (2017). The impact of rare variation on gene expression across tissues. Nature *550*, 239–243.

33. Rodgers, J.L., and Nicewater, W.A. (1988). Thirteen ways to look at the correlation coefficient. Am. Stat. *42*, 59–66.

34. Hamming, R.W. (1950). Error detecting and error correcting codes. Bell Syst. Tech. J. *29*, 147–160.

35. Climer, S., Templeton, A.R., and Zhang, W. (2014). Allele-specific network reveals combinatorial interaction that transcends small effects in psoriasis GWAS. Plos Comput. Biol. *10*, e1003766.

36. Climer, S., Yang, W., de las Fuentes, L., Dávila-Román, V.G., and Gu, C.C. (2014). A custom correlation coefficient (CCC) approach for fast identification of multi-SNP association patterns in genome-wide SNPs data. Genet. Epidemiol. *38*, 610–621.

37. Selkoe, D.J. (2011). Alzheimer's disease. Cold Spring Harb. Perspect. Biol. *3*, a004457.

38. Vassar, R. (2004). BACE1: the beta-secretase enzyme in Alzheimer's disease. J. Mol. Neurosci. *23*, 105–113.

39. Wongchitrat, P., Pakpian, N., Kitidee, K., Phopin, K., Dharmasaroja, P.A., and Govitrapong, P. (2019). Alterations in the expression of amyloid precursor protein cleaving enzymes mRNA in Alzheimer peripheral blood. Curr. Alzheimer Res. *16*, 29–38.

40. Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, S. (2009). Introduction to Algorithms, 3rd ed. (MIT Press).

41. Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. In Proceedings of the International AAAI Conference on Weblogs and Social Media, pp. 361–362.

42. Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. *13*, 2498–2504.

43. Fruchterman, T.M.J., and Reingold, E.M. (1991). Graph drawing by force-directed placement. Softw. Pract. Exp. *21*, 1129–1164.

44. Hu, Y. (2005). Efficient and high quality force-directed graph drawing. Math. J. *10*, 37–71.

45. Newman, M. (2018). Networks, 1st ed. (Oxford University Press).

46. Jain, A.K. (2010). Data Clustering: 50 Years beyond K-Means (Springer).

47. Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. Comput. J. *26*, 354–359.

48. Campello, R., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 160–172.

49. Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231.

50. Newman, M., and Girvan, M. (2004). Finding and evaluating community structure in networks. Phys. Rev. E *69*, 026113.

51. Ruan, J., and Zhang, W. (2008). Identifying network communities with a high resolution. Phys. Rev. E. Stat. Nonlin. Soft Matter Phys. *77*, 016104.

52. Duch, J., and Arenas, A. (2005). Community detection in complex networks using extremal optimization. Phys. Rev. E *72*, 027104.

53. Guimerà, R., and Nunes Amaral, L.A. (2005). Functional cartography of complex metabolic networks. Nature *433*, 895–900.

54. Clauset, A., Newman, M., and Moore, C. (2004). Finding community structure in very large networks. Phys. Rev. E *70*, 066111.

55. Newman, M.E.J. (2006). Modularity and community structure in networks. Proc. Natl. Acad. Sci. U S A *103*, 8577–8582.

56. Danon, L., Díaz-Guilera, A., Duch, J., and Arenas, A. (2005). Comparing community structure identification. J. Stat. Mech. Theor. Exp. *2005*, P09008.

57. Fortunato, S., and Barthélemy, M. (2007). Resolution limit in community detection. Proc. Natl. Acad. Sci. U S A *104*, 36–41.

58. Lea, J., and Climer, S. (2020). A search and filter strategy for identifying differentially co-expressed analyte modules. In Proceedings of the 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). https://doi.org/10.1109/BIBM49941.2020.9313573.

59. Karatzas, E., Gkonta, M., Hotova, J., Baltoumas, F.A., Kontou, P.I., Bobotsis, C.J., Bagos, P.G., and Pavlopoulos, G.A. (2021). VICTOR: a visual analytics web application for comparing cluster sets. Comput. Biol. Med. *135*, 104557.

60. Suárez-Fariñas, M., Lowes, M.A., Zaba, L.C., and Krueger, J.G. (2010). Evaluation of the psoriasis transcriptome across different studies by gene set enrichment analysis (GSEA). PLoS ONE *5*, e10247.

61. Huang, D.W., Sherman, B.T., and Lempicki, R.A. (2009). Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nat. Protoc. *4*, 44–57.

62. Zhou, Y., Zhou, B., Pache, L., Chang, M., Khodabakhshi, A.H., Tanaseichuk, O., Benner, C., and Chanda, S.K. (2019). Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. Nat. Commun. *10*, 1523.

63. Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. U S A *102*, 15545–15550.

Sharlee Climer is an assistant professor in the Department of Computer Science at the University of Missouri – St. Louis. She received her PhD from Washington University in St. Louis under fellowships from NDSEG and the Olin Fellowship, then continued as an NIH Postdoctoral Research Scholar at Washington University's School of Medicine. Her research focuses on combinatorial optimization with applications in genetics, with a primary focus on Alzheimer disease and COVID-19.