A Comparative Study of Gene Co-Expression Thresholding Algorithms

CARISSA BLEKER,1 STEPHEN K. GRADY,2 and MICHAEL A. LANGSTON2

ABSTRACT

The thresholding problem is studied in the context of graph theoretical analysis of gene co-expression data. A number of thresholding methodologies are described, implemented, and tested over a large collection of graphs derived from real high-throughput biological data. Comparative results are presented and discussed.

Keywords: biological data analysis, graph theoretical algorithms, thresholding.

1. INTRODUCTION

FINITE, SIMPLE, UNDIRECTED GRAPHS are often used in biological data analysis. In this context, a vertex is typically used to denote a gene, protein, metabolite, or some other biological entity. An edge between two vertices represents some relationship between them and is typically weighted by an association metric computed over its endpoints. Thresholding is then applied to discard weak, dubious, or unconvincing relationships so that subsequent analysis can concentrate on the most significant experimental features of interest.

Various graph theoretical strategies have been proposed in an effort to determine a most appropriate threshold, but no overall consensus has been reached as to which techniques are best suited to various graph classes (Bleker, 2020). Given the vast abundance of gene–gene correlation data available and their use in innumerable life science studies, the class on which we focus here is the gene co-expression graph (sometimes called a gene co-expression network). The primary aims of this comprehensive study are to survey, realize, and in some cases improve upon existing thresholding techniques and to compare them systematically over an extensive testbed of representative data.

In Section 2, we briefly introduce graph theoretical notation and discuss the foundational significance of correlation. In Section 3, we describe and categorize current state-of-the-art thresholding strategies, to which we add a few enhancements in Section 4. In Section 5, we discuss the biological data, algorithms, and implementations we compiled for systematic methodological evaluations. In Section 6, we present and review empirical results. And in Section 7, we draw a few conclusions from this work, posit directions for future research, and describe a pair of publicly available repositories we created so that the research community can access and experiment with the thresholding algorithms under study.

1Department of Biotechnology and Systems Biology, National Institute of Biology, Ljubljana, Slovenia.
2Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, Tennessee, USA.
2. CORRELATION AS AN ORGANIZING PRINCIPLE

Let the ordered triple \( G = (V, E, W) \) denote a graph with vertex set \( V \), edge set \( E \), and edge weight set \( W \). The weight \( w_{i,j} \) assigned to edge \( e_{i,j} \) is critical in that it represents some measure of similarity between its endpoints \( v_i \) and \( v_j \). Correlation is generally used for this purpose across a diverse plethora of scientific applications, making it a unifying theme and a key organizing principle in the graph theoretical analysis of biological and numerous other sorts of data. Established correlation metrics include the Pearson coefficient (denoted by \( r \)), which is undoubtedly the most widely used choice even despite its assumptions of linearity. Other well-known correlation metrics include rank measures such as Spearman’s rho and Kendall’s tau. Meanwhile, the Jaccard index has proven useful for set theoretic comparisons, while cosine similarity has become popular for textual data. Mutual information, the Czekanowski index, and a wide variety of other similarity formulae have been proposed, applied, and studied.

3. POPULAR THRESHOLDING APPROACHES

The majority of thresholding algorithms can be classified as “hard” in the sense that an edge is retained if and only if its weight exceeds some predetermined cutoff value. Depending on the application, vertices that become isolated as a result of edge deletions may be removed. When such an operation is applied uniformly across all edges, it is naturally called “global” thresholding. In contrast, the process is termed “local” when neighborhoods of the graph are thresholded differently from others (Guzzi et al., 2014). Local methods may be useful in the presence of local signal or noise, but they may also introduce local bias.

Hard thresholding approaches can be further subdivided into those that utilize graph structure versus those that employ statistical modeling and depend on underlying assumptions about data distributions. A number of other miscellaneous thresholding schemes exist as well. As a rule of thumb, too low a threshold may generate false positives, while too high a threshold runs the risk of false negatives. An alternate thresholding approach is termed “soft” in the sense that edge weights may be transformed to suit the threshold. Such a method may, for example, apply a (typically sigmoidal) function across all edge weights so that relatively low edge weights are further reduced while relatively high edge weights are further increased (Zhang and Horvath, 2005). We mention soft thresholding only in the spirit of completeness and do not consider it in this comparative analysis.

3.1. Thresholding based on graph structure

Foreknowledge of graph structure can sometimes dictate the choice and application of thresholding methodology, and a variety of thresholding techniques can be viewed in this context. Representative examples include algorithms based on scale-free graph models, those that employ spectral graph theory or random matrix theory, and those that focus on cliques, clustering coefficients, or other measures of graph density. We refer the interested reader to our recent preliminary work as described in Bleker et al. (2023), where we study these and other structural approaches to the omics thresholding problem.

3.2. Thresholding based on statistical criteria

Parametric statistical methods assume data that can be modeled by probability distributions with known preset parameters. In contrast, nonparametric methods make no such assumptions. One must take care with either approach, however, because overreliance on \( p \) values is well known, and a statistically significant edge weight may actually represent only weak biological signal. For example, a large number of samples will almost always result in low \( p \) values, while the magnitude of the effect or correlation could be very small (Sullivan and Feinn, 2012).

3.2.1. Relevance networks. This problem has been studied in the context of transcriptomic correlation graphs produced under the well-known relevance network model (Allocco et al., 2004). Using microarray data from more than 600 yeast hybridization experiments, it was found that genes were more than 50% likely to share a common transcription factor (and not merely be co-expressed) when the absolute value of their Pearson’s correlation coefficient was at least 0.84. Similar analyses have been used to set
thresholds from 0.8 (Sanoudou et al., 2003) to 0.875 (Voy et al., 2006). In general, this approach requires considerable prior knowledge of the data under investigation, and the selection of a threshold that supports this knowledge.

3.2.2. Significance and power. False positives in the edges can be controlled by only keeping edges that correspond to a $p$ value smaller than a set significance value. An example use of this method is in Lee et al. (2004), where they use Bonferroni-corrected Pearson’s correlation $p$ values, and control the error rate at 0.01.

In contrast, one can also control the power of the edges. Such an approach is carried out in Borate et al. (2009). For a number of values of the Pearson correlation, the power is calculated using significance of 0.05. The correlation that corresponded to a power of at least 0.80 was used as the threshold.

When we cannot make an assumption on the distribution of the correlations, we can estimate it from empirical distributions derived from permutations or bootstrapping.

3.2.3. Bayesian networks. Bayesian networks provide an analytical framework for identifying the most statistically likely structure of conditionality among variables. A thresholding method based on such networks, using bootstrapping and model averaging, was developed in Scutari and Nagarajan (2013) in an effort to identify edges of significance in molecular networks. Under current technologies, however, Bayesian methods have been applied to data sets with relatively few vertices (variables) and require large numbers of observations to learn (“large $n$, small $p$ problems”; Scutari et al., 2019). In contrast, biological network analytics generally demand the ability to handle large numbers of variables, with limited observations (“small $n$, large $p$”), thereby limiting the utility of Bayesian thresholding applications at any sort of big data level.

3.3. Ad hoc thresholding approaches

A handful of thresholding strategies have been proposed that rely on arbitrary decisions, making use of neither graph structure nor statistical relevance. Thus, they can be somewhat difficult to classify (and, debatably, rather difficult to justify as well).

3.3.1. Edge rank. In Ruan et al. (2010), a method is proposed that retains only the $d$ most highly weighted edges adjacent to each vertex, with $d$ set somewhere between 3 and 5. It is shown that this usually results in a connected graph in both synthetic and real-world data. While there is no statistical justification for such an approach, it does tend to produce graphs with wide a spectrum of edge weights.

3.3.2. Preset value. One may also select a threshold subjectively based on preconceived criteria. In transcriptomic network analysis, for example, preset Pearson’s correlation thresholds of 0.70 and 0.90 were employed in Freeman et al. (2007), while a cutoff of 0.60 was used in Zhou et al. (2002).

3.3.3. Fixed percentile. It can be intuitively appealing to retain only some fixed predetermined percentage of edges. This again is an arbitrary choice, however, and one with neither provable justification nor formal basis in biological significance. In Pellegrino et al. (2004), for instance, the highest 1% of correlations were naturally found to be the most enriched in orthologous pairs of expressed sequence tag clones, and based on this, Ala et al. (2008) used the top 1% of correlations as edges in co-expression graphs. As another example, in Lee et al. (2004), only edges with weights in the top and bottom 0.5% were retained.

4. ALGORITHMIC ADVANCES

We implemented a number of these thresholding algorithms, devised several methodological improvements, and designed a few previously unreported thresholding techniques amenable to systematic comparisons. We refer the curious reader to our recent preliminary work as reported in Bleker et al. (2023), where we describe stability and efficiency enhancements for algorithms based on scale-free models, maximal clique, and global-local measures.
We also studied the application of percolation metrics. Percolation theory (not to be confused with clique percolation) refers to the study of fluid flow through porous media and can be used by way of analogy to denote a measure of graph connectivity. If a graph is well connected, then information can easily flow (percolate) from one vertex to another. In the present context, percolation is defined as the process of randomly and independently removing edges of the graph with a given probability. For most random graphs, there exists a critical probability, the percolation threshold, that results in a giant connected component that is orders of magnitude larger than the smaller components in the graph. The notion of percolation prompts a variety of possible thresholding approaches in nonrandom graphs. For example, we can increase the threshold until just before the graph contains multiple connected components. We term this the single-component threshold. (Ironically, a similar definition is named the percolation threshold in Bullmore and Bassett, 2011.) Alternately, we may produce a somewhat higher threshold by permitting multiple connected components and stopping the threshold increase just before the size of the largest component begins to falter. We call this the cc-inflection threshold. And finally, we can allow the increase to continue until just before any vertices are isolated. We dub this the whole-graph threshold.

5. IMPLEMENTATION AND TESTING

We now discuss our analytical milieu, the thresholding methods we chose for evaluation, our benchmark suite of test cases, and the experimental procedures we employed. Each thresholding algorithm we have thus far described is reasonably fast and straightforward to compute. In this analysis, we are not, therefore, particularly concerned with relative efficiencies, competitive timings, or high-performance implementations. Instead, we are mainly focused on thresholding solution quality, and so we employed an assortment of hardware platforms and applications software based primarily on availability. We refer the interested reader to Bleker et al. (2023) for detailed descriptions of these various support systems.

5.1. Thresholding methods tested

After careful scrutiny, we selected 15 of these algorithms for evaluation and testing. In so doing, we sought methods that were data dependent (eliminating relevance networks), computationally scalable (eliminating Bayesian networks and local-global methods), and rigorously grounded (eliminating edge rank, preset value, and fixed percentile) (Table 1). Four of these techniques (indicated by shaded rows) we developed for this effort. The remaining 11 we realized as detailed and/or improved in the preceding algorithmic descriptions.

5.2. Raw data testbed

We crafted a benchmark test suite of 83 graphs from an assortment of nonsynthetic biological data sources. Vertices were used to denote genes/transcripts, with each edge weighted by the Pearson correlation coefficient of its endpoints. Although gene co-expression is often studied using only transcriptomic data, DNA methylation data can be revealing as well because it generally correlates inversely with transcription and is known to play a key role in mediating gene expression (Suzuki and Bird, 2008). Accordingly, the following data repositories were used:

- EntropyExplorer (Wang et al., 2015), from which we obtained 19 graphs using transcriptomic disease case–control microarray data,
- Genomic Data Commons (Grossman et al., 2016), from which we obtained 44 graphs using DNA methylation β values of cancer case–control cohorts, and
- ManyMicroarray (Lee et al., 2004, 2019), from which we obtained 20 graphs using prenormalized human microarray data.

A list of all files used in our test suite can be found in Supplementary Table S1.

5.3. Experimental design

We began with an initial threshold of 0.10 for each graph. Occasionally, we needed to reduce a graph’s size to satisfy memory or file transfer limitations, which we accomplished by increasing its threshold in
units of 0.10 until its size was at most 3GB. Each of our 15 thresholding strategies was then invoked. In a few cases, algorithms exceeded resource limits or failed to converge. Overall, 991 thresholds were thereby identified. Those below 0.60 were discarded, leaving 644 thresholds for continued analysis.

A floor of 0.60 was deemed suitable for several reasons. First off, downstream analysis can be prohibitively time-consuming at lower cutoffs, and yet it is known that transcriptomic data can demand thresholds upward of 0.84 to produce a true positive association rate of merely 50% (Allocco et al., 2004). In addition, we found the median threshold produced across all algorithms under study to be 0.68, while methods purportedly developed specifically for transcriptomic data all yielded thresholds between 0.80 and 0.96.

We next performed clustering on these 644 thresholded graphs, using the paraclique algorithm (Chesler and Langston, 2005), a leading-edge graph theoretical technique previously shown to generate clusters of measurably superior quality (Jay et al., 2012). Paraclique was invoked with glom term one (Hagan et al., 2016) and allowed to produce at most 300 clusters per thresholded graph, with 11 runs that were halted at 60 cpu hours. This resulted in 633 clustered graphs.

In the end, we were left with 64,373 clusters distributed across these 633 graphs. Enrichment analysis based on the Gene Ontology (GO) project (Ashburner et al., 2000) was employed to estimate biological fidelity via the Panther web API (Mi et al., 2017) and a custom Python script. A cluster was marked as significant if and only if its contents included one or more significantly overrepresented GO terms.

6. EMPIRICAL RESULTS

In Figure 1, we display threshold ranges as computed by each method we investigated. A handful of these may merit discussion. Cluster-Separation, for example, found no thresholds for most of the
EntropyExplorer and Genomic Data Commons inputs, and only exceptionally high thresholds for the ManyMicroarray data. Quite possibly this method was hampered by its need for skewed edge weight distributions. Maximal clique ratio-based (MCR-based) methods and Scale-Free also delivered exceedingly high thresholds, presumably because of MCR’s reliance on perturbations in graph density and Scale-Free’s inherent dependence on graph topology. In contrast, Power and Type I returned thresholds so low that they seldom exceeded our 0.60 lower bound. Perhaps this was due to their need for huge numbers of inputs. And finally, MCR-2 and MCR-3 sometimes took a relatively long time to terminate, likely due to their stringent clique doubling and tripling requirements.

FIG. 1. Thresholds produced by data source. (A) Entropy Explorer, (B) GDC, and (C) ManyMicroarray. GDC, Genomic Data Commons.
In Table 2, we report results from GO cluster annotation. There we list four values for each algorithm, namely the number of graphs for which a suitable threshold was determined, the number of gene clusters generated, the number of these clusters that were significant, and a novel metric we dub the total significant clusters ratio (TSCR). We computed this metric by adding the individual significance ratios of the clusters found within each graph. TSCR’s simple design is aimed to ensure that no single graph can overwhelm the analysis and to produce a figure of merit that does not directly depend on the number of clusters produced. We illustrate its use with the Cluster-Separation algorithm. This method generated clusters from five graphs. In one graph, only one cluster was found, and it was significant (enriched). In a second graph, two clusters were found of which only one was significant. In a third graph, 7 out of 10 clusters were significant, and so on. Thus, the Cluster-Separation TSCR was:

\[ \frac{1}{1} + \frac{1}{2} + \frac{7}{10} + \frac{1}{1} + \frac{4}{5} = 4. \]

An exhaustive list of TSCR calculations is provided in Supplementary Table S2.

Algorithms proposed specifically for transcriptomic data performed admirably, with the five best-scoring methods (the three MCR-based strategies, Spectral-Methods, and Density-Inflection) each based on fluctuations in graph density. Curiously, the next highest scoring method (Scale-Free) relies instead on graph sparsity (Del Genio et al., 2011), which probably prompted it to return its aforementioned high thresholds.

### 7. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Thresholding is a crucial preliminary operation in a wide spectrum of graph theoretical applications. Here, we have discussed, implemented, and rigorously tested a collection of thresholding methods proposed for high-throughput biological data analysis. At the same time, we formulated a few new thresholding techniques, constructed a benchmark suite of relevant data, and suggested a straightforward performance metric in an effort to evaluate algorithmic behavior. The scope of this effort brings with it a few limitations, most notably its restriction to transcriptomics and its reliance on ontological enrichment as a measure of cluster quality.

#### 7.1. Moving forward

A similar analysis could probably be performed on proteomic, genomic, metabolomic, and other sorts of omic data, although it is not clear how best to rate thresholding effectiveness without something akin to ontological enrichment. And we would expect substantially different relative algorithmic performance, due in large part to variations in topological graph characteristics. While gene co-expression data can be described by transcription factors and other gene regulatory mechanisms that tend to produce overlapping clusters and hub-spoke features, we know from experience that proteomic data, for example, are typically defined by protein complexes that produce disjoint subgraphs and nonoverlapping cliques.
The downstream effects of noise may also merit continued investigation. This is because noise often remains a concern in high-throughput biological data, and it is in fact one reason we chose to use the paraclique algorithm for clustering. Thus, it might be worth checking on thresholding robustness and repeatability (Lu et al., 2019), say, by conducting a series of tests that systematically introduce noise. In the same vein, one could conduct a more fine-grained approach and employ a range of glom term parameters (Hagan et al., 2016) to learn how more aggressive noise abatement strategies can affect TSCR scoring. In a study of type 1 diabetes mellitus, for example, bounds on paraclique density led us to use a glom term of five (Eblen et al., 2009). Of course this sort of scheme would be cumbersome, time-consuming, and tedious on a large scale because it would require individual tuning for each data set in a benchmark test suite.

7.2. A thresholding resource

At https://github.com/carissableker/thresholding and https://zenodo.org/records/10532019, we created a pair of publicly available repositories so that researchers can experiment with various thresholding methodologies, compute relevant statistics, identify promising thresholds for further study, and even reproduce our work. The former is a GitHub site that provides the user with instant access to all 15 of the thresholding methods we implemented. The latter is a Zenodo site containing all 83 of the graphs we compiled for our benchmark test suite. For these, we employed the igraph C library (Csardi and Nepusz, 2006) for graph theoretical analysis and the ALGLIB library (Bochkanov, 2019) for statistical distributions. These routines were instrumental in our creation of this resource. For ease of use, input graph formatting has been simplified to a single list of edge weights. Thresholds may be computed using either signed or absolute values of these weights as desired. A thresholding repository such as this has not, to the best of our knowledge, previously been curated and released to the scientific community.

AUTHORS’ CONTRIBUTIONS


ACKNOWLEDGMENTS

A preliminary version of a portion of this article was presented at the 19th International Symposium on Bioinformatics Research and Applications held in Wroclaw, Poland, in October 2023.

AUTHOR DISCLOSURE STATEMENT

The authors declare they have no conflicting financial interests.

FUNDING INFORMATION

This research was supported in part by the National Institutes of Health under grant R01HD092653 and by the Environmental Protection Agency under grant G17D112354237. It also employed resources of the National Energy Research Scientific Computing Center, a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory and operated under Contract No. DE-AC02-05CH11231.

SUPPLEMENTARY MATERIAL

Supplementary Table S1
Supplementary Table S2
REFERENCES


Eblen JD, Gerling IC, Saxton AM, et al. Graph algorithms for integrated biological analysis, with applications to type 1 diabetes data. In: Clustering Challenges in Biological Networks. World Scientific Publishing Company; 2009; chapter 9; pp. 207–222.


Address correspondence to:
Dr. Stephen K. Grady
Department of Electrical Engineering and Computer Science
University of Tennessee
Knoxville, TN 37996
USA

E-mail: sgrady3@vols.utk.edu