An Algorithm for Identifying Subnetworks Preserved Across Datasets

The `cmdnet` package for R

Nicholas Wisniewski
University of California, Los Angeles

October 17, 2016

Abstract

Interest in integrating multiple public genomics datasets is growing as more scientific data is shared. Network analysis tools are important in analysis of high-dimensional data such as genomics, but difficulties arise when faced with multiple sources of data due to large variability across datasets. Here we present an algorithm, `cmdnet`, that attempts to understand this variability by computing correlation matrix distance (CMD) statistics for an ensemble of networks. It can be used to quantify overall and node-specific differences between network datasets, and rank nodes by consistency of edge weights. After deriving the main statistics, we show an example usage, and demonstrate how to identify preserved subnetworks and reduce noise by removing highly variable nodes. Finally, we use `cmdnet` in conjunction with WGCNA to show that preserved subnetworks are unbiased with respect to modular structure, being evenly distributed across the WGNCA network. We also show that a dataset can be filtered significantly while still recovering the modular structure found by full-network reconstruction.

Contents

1 Introduction

The `cmdnet` algorithm computes correlation matrix distances (CMD) for an ensemble of networks, which can be used to quantify variability across networks, and as a noise reduction tool to reduce variability. This application is based on networks that can be represented by correlation matrices, though alternative measures of undirected adjacency (e.g. MIC, mutual information) are also compatible (using the `corFUN` option). The algorithm works by first computing an average or consensus network, and then quantifying variability by measuring deviations from the mean.

The distance measures used by `cmdnet` to quantify deviations are based on the angular distance between vectors, or what is often referred to as cosine similarity. This measure arises from the familiar Euclidean dot product formula, relating the inner product of two vectors a and b to the cosine of the subtended angle,

$$\langle a, b\rangle = \|a\| \, \|b\| \, \cos\theta.$$

Consequently, $\cos\theta$ is generally used to measure similarity between two vectors, since $\cos\theta \to 1$ when vectors are identical, $\cos\theta \to 0$ when vectors are orthogonal, and $\cos\theta \to -1$ when vectors are anti-parallel. This relationship provides a computationally efficient way to compute similarity in very large datasets, such as when inferring a network from data. Indeed, the Pearson correlation coefficient can be understood as an application of this inner product.

It is also necessary to work with measures of distance, rather than similarity, when computing statistics like deviations. For these purposes, it is common to simply flip the unit interval so that similar vectors tend to zero, while orthogonal vectors tend to one,

$$d = 1 - \cos\theta.$$

However, this measure is not a distance measure in the strictest sense, as it does not satisfy the triangle equality. To recover a true distance measure, we can instead solve for the angle θ directly,

$$\theta = \cos^{-1}\left(\frac{\langle a, b\rangle}{\|a\| \, \|b\|}\right),$$

and then normalize it to get a value between zero and one. The normalization chosen depends on whether the inner product is strictly positive,

$$d_\theta = \frac{2}{\pi}\cos^{-1}\left(\frac{\langle a, b\rangle}{\|a\| \, \|b\|}\right), \qquad 1 \geq \langle a, b\rangle \geq 0$$

$$d_\theta = \frac{1}{\pi}\cos^{-1}\left(\frac{\langle a, b\rangle}{\|a\| \, \|b\|}\right), \qquad 1 \geq \langle a, b\rangle \geq -1$$

Below, we generalize this distance measure to sets of correlation matrices to provide fast measures of network differences across datasets.

1.1 Node-Specific Deviations

A node in a graph can be described by a vector of edge weights encoding its connections. When a network is represented by a correlation matrix ρ, the vector of edge weights for node j corresponds to the j-th row of the correlation matrix, $\rho_{j\cdot}$. Similarity between a node's connectivity in two networks $\rho_{1,j\cdot}$ and $\rho_{2,j\cdot}$ can then be summarized by the cosine similarity explained above. This leads directly to an angular distance measure,

$$d_\theta(\rho_{1,j\cdot}, \rho_{2,j\cdot}) = \frac{1}{\pi} \cos^{-1}\left(\frac{\langle \rho_{1,j\cdot}, \rho_{2,j\cdot} \rangle}{\|\rho_{1,j\cdot}\| \, \|\rho_{2,j\cdot}\|} \right).$$

With this as a distance metric between two networks, we can proceed to derive a measure of variability over several networks. Using the set of correlation matrices, we compute an average network $\bar{\rho}$, and then measure variability by computing the deviations $\delta_{i,j}$ of each node j in network i away from the average,

$$\delta_{i,j} = d_\theta(\rho_{i,j\cdot}, \bar{\rho}_{j\cdot}); \quad \bar{\rho} = \sum_{i=1}^{m} w_i \, \rho_i; \quad |w| = 1.$$

We then calculate the mean deviation $\bar{\delta}_j$ for each node, providing a scalar measure of variance in node connectivity across m datasets,

$$\bar{\delta}_j = \sum_{i=1}^{k} w_i \, \delta_{i,j}.$$

It follows that subsets of nodes with low mean deviation describe subnetworks that are well preserved across datasets, suggesting a fast technique for noise reduction in network analysis. Next, we create a metric to measure the reduction in noise, by quantifying overall network similarity.

1.2 Correlation Matrix Deviations

The notion of cosine similarity can be generalized to matrices in order to measure the distance between two networks. This generalization is known as the correlation matrix distance (CMD), which we write here in terms of the angular distance,

$$D_\theta(\rho_1, \rho_2) = \frac{2}{\pi} \cos^{-1}\left(\frac{\mathrm{tr}\{\rho_1 \rho_2\}}{\|\rho_1\| \, \|\rho_2\|} \right).$$

This distance is 0 if two matrices are equal up to a scaling factor, and 1 when they are maximally different. It is an attractive metric because because it can be vectorized, viewing orthogonality between $n \times n$ matrices as orthogonality in an n^2 dimensional vector space.

With this as a measure of distance between networks, we derive a measure of variability over a set of networks, analogous to the node-specific measure. Using the average network $\bar{\rho}$ computed earlier, we measure variability by taking the deviations Δ_i from the average for each network i,,

$$\Delta_i = D_\theta(\rho_i, \bar{\rho}); \quad \bar{\rho} = \sum_{i=1}^{m} w_i \, \rho_i; \quad |w| = 1.$$

We then calculate the mean deviation $\bar{\Delta}$, providing a scalar measure of variance in network connectivity across m datasets

$$\bar{\Delta} = \sum_{i=1}^{m} w_i \, \Delta_i.$$

Successful noise reduction schemes should minimize this quantity. In our results, we show how selecting subnetworks based on the node-specific deviations can be used to filter out network noise.

1.2.1 Cosine Dissimilarity

Here we note that the alternative distance measures based on cosine similarity, which do not calculate the angle directly,

$$d(\rho_{1,j\cdot}, \rho_{2,j\cdot}) = 1 - \frac{\langle \rho_{1,j\cdot}, \rho_{2,j\cdot} \rangle}{\|\rho_{1,j\cdot}\| \, \|\rho_{2,j\cdot}\|},$$

$$D(\rho_1, \rho_2) = 1 - \frac{\mathrm{tr}\{\rho_1 \rho_2\}}{\|\rho_1\| \, \|\rho_2\|},$$

are also available in `cmdnet` and can be accessed by setting the option `metric = "cosine"`.

2 Preliminaries

The `cmdnet` package can be found on GitHub, and installed with the `devtools` package by typing: `install_github("nwisn/cmdnet")`. There is only one necessary input, and that is a list of data arrays in usual format (dataframe or matrix), where each row is a node, and each column is an independent measurement (e.g. expression level). In our example, this list is named `datalist`. There is a second optional input, which is a vector of nodes-of-interest. In our example, this vector is named `geneset`, and we use it to limit our analysis to only 268 genes.

Gene array datasets generally will have different numbers of genes, and the ordering of the genes in the data arrays are different across datasets. To handle this automatically, `cmdnet` has a function named `cleandDatalist` to perform the matching, and select only the genes that occur in every dataset.

3 Running `cmdnet`

The `cmdnet` function runs in three general modes, which can be computed together or separately, and specified by the option `mode = c("cmd","row","net")`. They are as follows:

- The ‘"cmd"‘ mode calculates the mean correlation matrix deviation $\bar{\Delta}$ by computing the distance Δ_i between the i-th network and the averaged network $\bar{\rho}$, and then averaging over all networks according to sample size.
- The ‘"node"‘ mode calculates node-specific correlation deviations $\delta_{i,j}$, and the mean deviations $\bar{\delta}_j$ describing the variability of each node j's connectivity across datasets.

- The ‘"net"‘ mode computes the correlation matrix distance $D_{ij} = D(\rho_i, \rho_j)$ between each pair of networks, enabling clustering of networks in the ensemble, and identification of outlier networks.

By default, all three modes of `cmdnet` are computed, but any subset can be specified in the function call using the `mode` option. Other options will be detailed later.

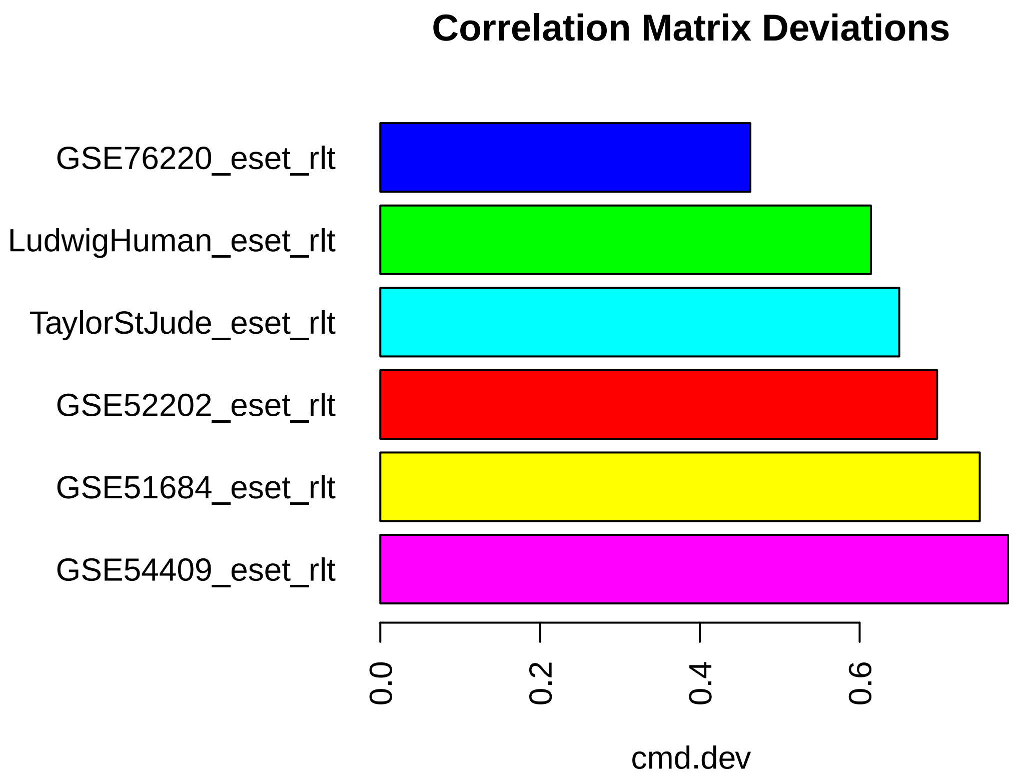
```
result <- cmdnet(datalist, geneset, mode=c("cmd","node","net"))
names(result)
```

```
##  [1] "cmd.dev"       "cmd.meandev"   "node.dev"      "node.meandev"
##  [5] "node.dev.cor1" "node.dev.cor2" "net"           "group.names"
##  [9] "group.r"       "group.n"       "total.n"       "r0"
```

The output of `cmdnet` is a list of 12 statistics. If all modes are selected, then all the statistics will be computed, and every element of the list will be filled. However, if only a subset of modes were selected, then some elements will be null. In addition to the statistics resulting from each mode, the result also contains statistics about each group, such as the names of each dataset or group `group.names`, the correlation matrices `group.r` of each dataset, the sample size of each dataset `group.n`, the total sample size `total.n`, and the mean correlation matrix `r0`.

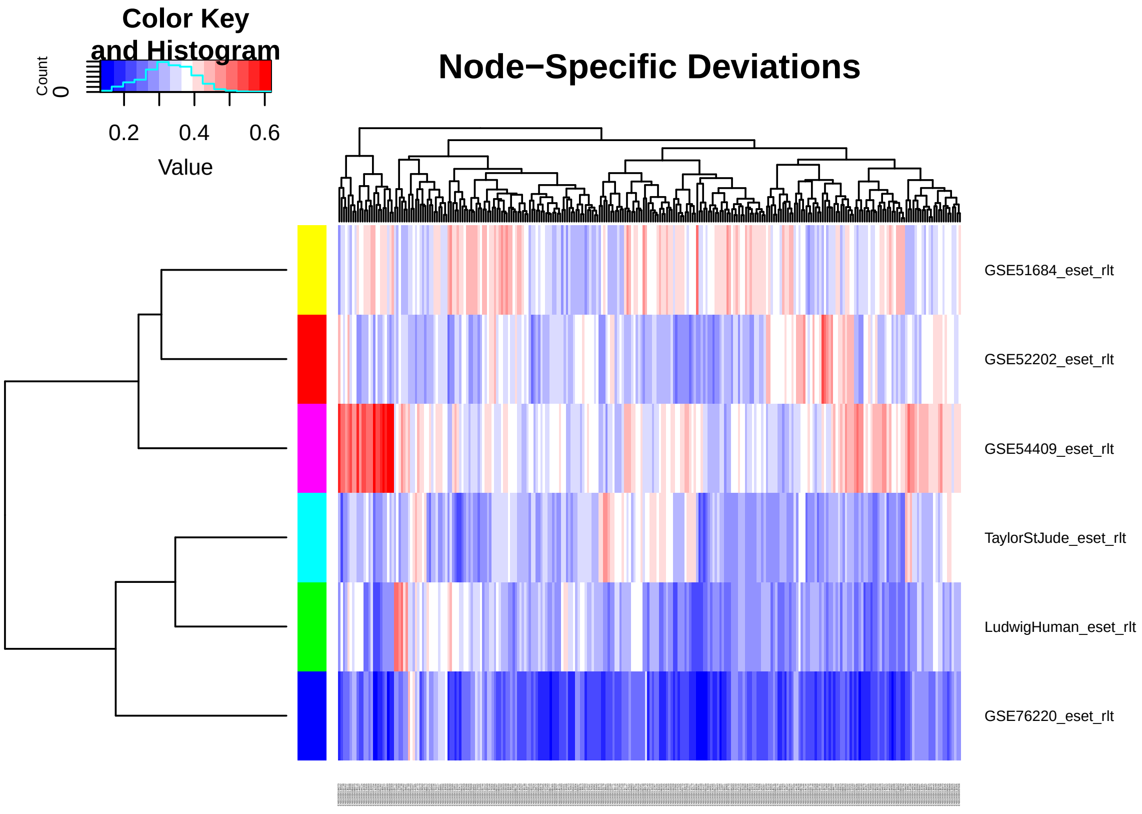
3.1 Correlation Matrix Deviations

We computed the deviations Δ_i; there is one deviation statistic for each dataset. Outlier datasets can be identified by looking for extreme deviations. We notice that the blue dataset is closest to the average network, and the magenta dataset is farthest. It's worth noting that the magenta dataset has extremely small sample size ($n = 5$), and is therefore suspect. Later when we compute node-specific, we can see if there are only certain genes that are problematic, or if the whole dataset should be dropped.



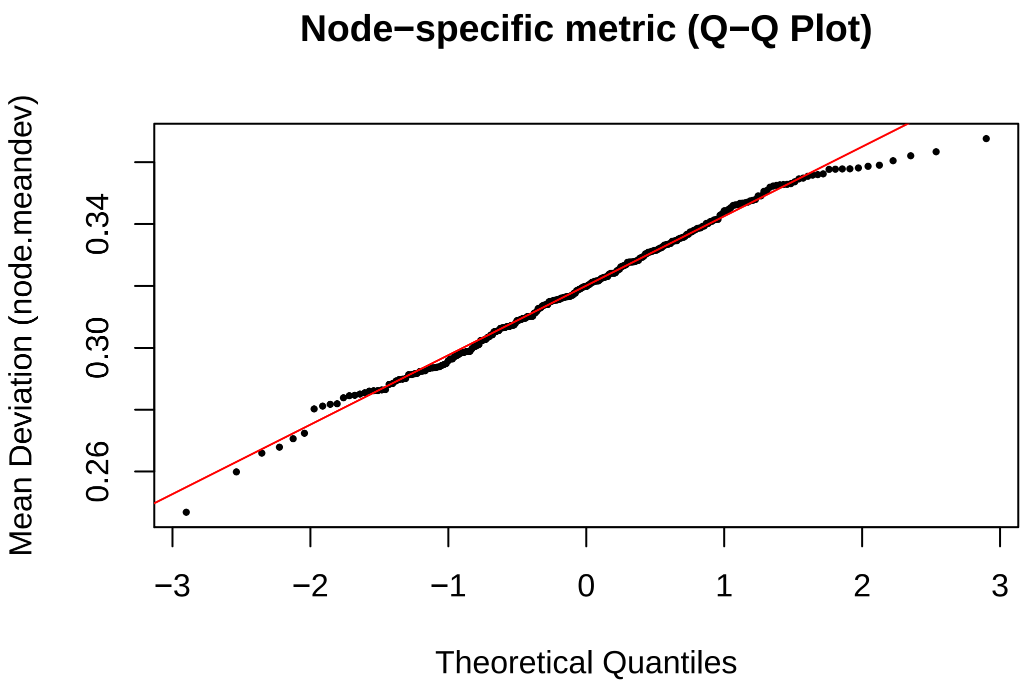
3.2 Node-Specific Deviations

We computed deviations $\delta_{i,j}$ for each of the nodes in each dataset from the mean network (`node.dev`). We notice that the blue dataset has much smaller deviations than the others. This again shows that it is most similar to the average network. There is also a cluster of genes in the magenta dataset that have very large deviations; this is the dataset with extremely small sample size ($n = 5$). This suggests that maybe only one cluster of genes is highly suspect in the magenta dataset, and that the dataset can be salvaged by removal of these genes.



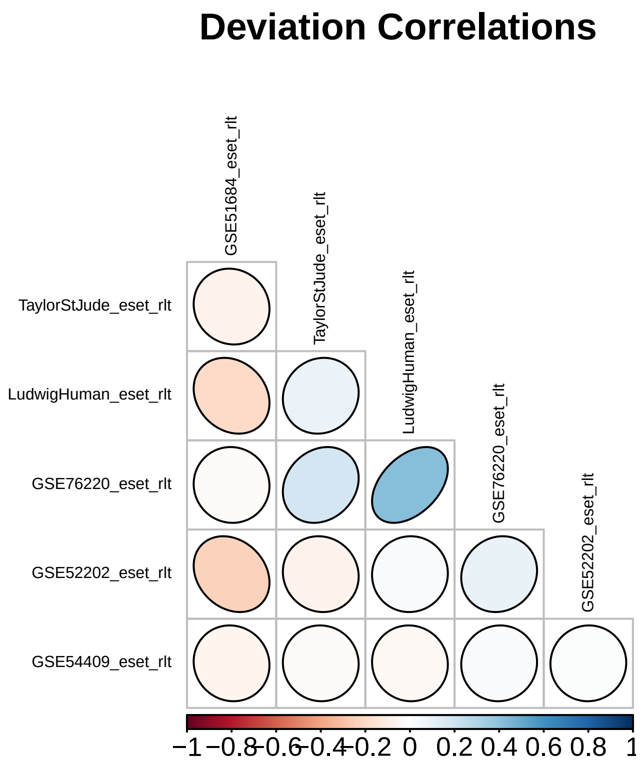
3.3 Node-Specific Mean Deviations

The mean deviations $\bar{\delta}_j$ are computed by weighted average of the node deviations $\delta_{i,j}$ across all datasets (`node.meandev`). Smaller mean deviations indicate better preservation across datasets, while larger deviations indicate changes in connectivity across datasets. These deviations can act as a ranking scheme for identifying the most preserved relationships across datasets. We observed that the are approximately normally distributed.



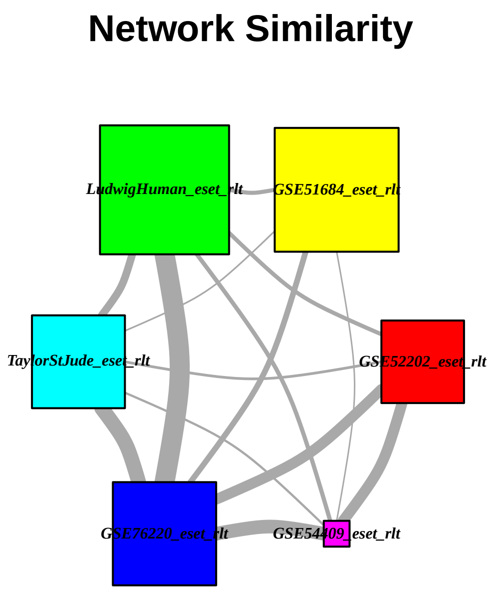
3.4 Node-Specific Deviation Correlations

We computed correlations between deviations in the datasets by looking across nodes (`node.dev.cor1`). We expect these deviations should be largely uncorrelated, and that correlations may indicate hidden biases among datasets. Here, we detect a moderate correlation (r=0.43) between two of the datasets, which we cannot explain.



3.5 Dataset Similarity Network

The "net" mode computes pairwise correlation matrix distances between networks, so that the datasets themselves may be viewed as a network. It returns only 1 statistic, `net`, containing the network similarity matrix. We depict the datasets as a network using the similarity matrix. The size of each dataset is proportional to the sample size.



4 Noise Reduction

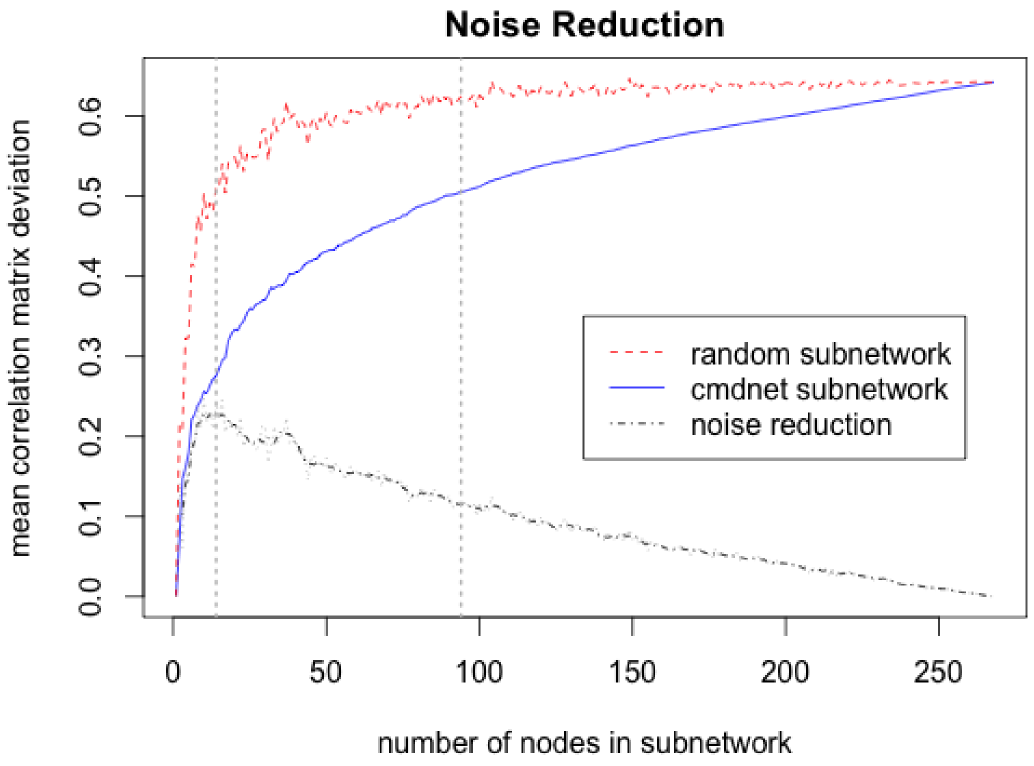
The `cmdnet` algorithm can be used to reduce noise across datasets by identifying subnetworks that are maximally preserved. Here we show that reduction in noise is possible by optimizing a subnetwork of smaller size, thereby filtering out the least consistent genes.

4.1 Post-Network Reconstruction

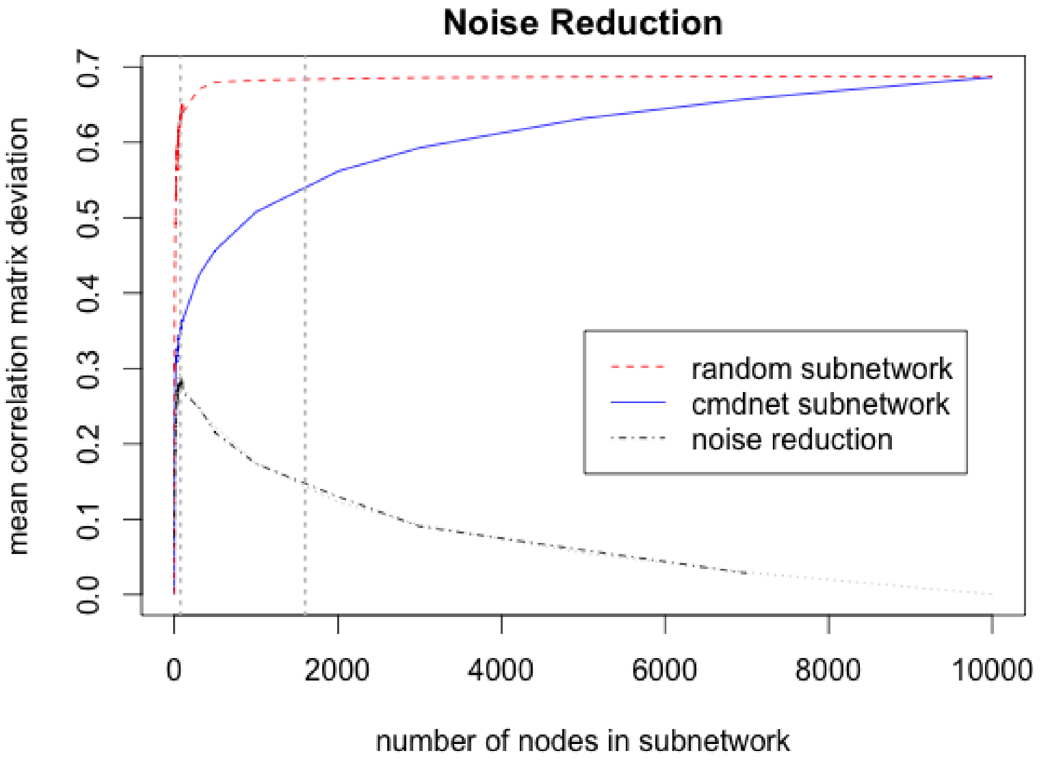
It is common practice to use correlation network analysis algorithms (e.g. WGCNA, MICA) to cluster full genomic datasets, compute module eigengenes, and finally identify “disease modules” related to phenotypes or traits. However, after a disease module is found, the process of understanding the module remains a difficult process that involves creative application of many bioinformatics tools, and which can still be complicated by the presence of hundreds or thousands of genes. In order to isolate even smaller gene sets, a variety of metrics or scoring systems are usually developed to indicate some property. For example, correlation with the phenotype or trait creates a kind of priority score for considering predictive genes. Additionally, metrics computed from the correlation network itself are also commonly used, such as degree connectivity or correlation with the module eigengene.

Here we propose a priority score based on the mean correlation deviation `node.meandev` for each node. Nodes with low mean correlation deviation are to be considered most reproducible when viewed across different datasets. It stands to reason that if we choose a subnetwork consisting of only nodes with the smallest mean correlation deviation, that subnetwork will be most reproducible.

Below, we demonstrate how this approach can be used to effectively reduce noise in network analysis across datasets. First, we restrict our analysis to a small gene list (268 genes) corresponding to a previously identified disease module. We build subnetworks of various sizes using this criteria, and compute the subnetwork’s mean correlation matrix deviation across datasets. To compute the effective noise reduction, we subtract this from the mean correlation matrix deviation found by creating a random subnetwork. We note that maximum noise reduction occurs for a set of about 15 nodes, and half-max at about 95 nodes.

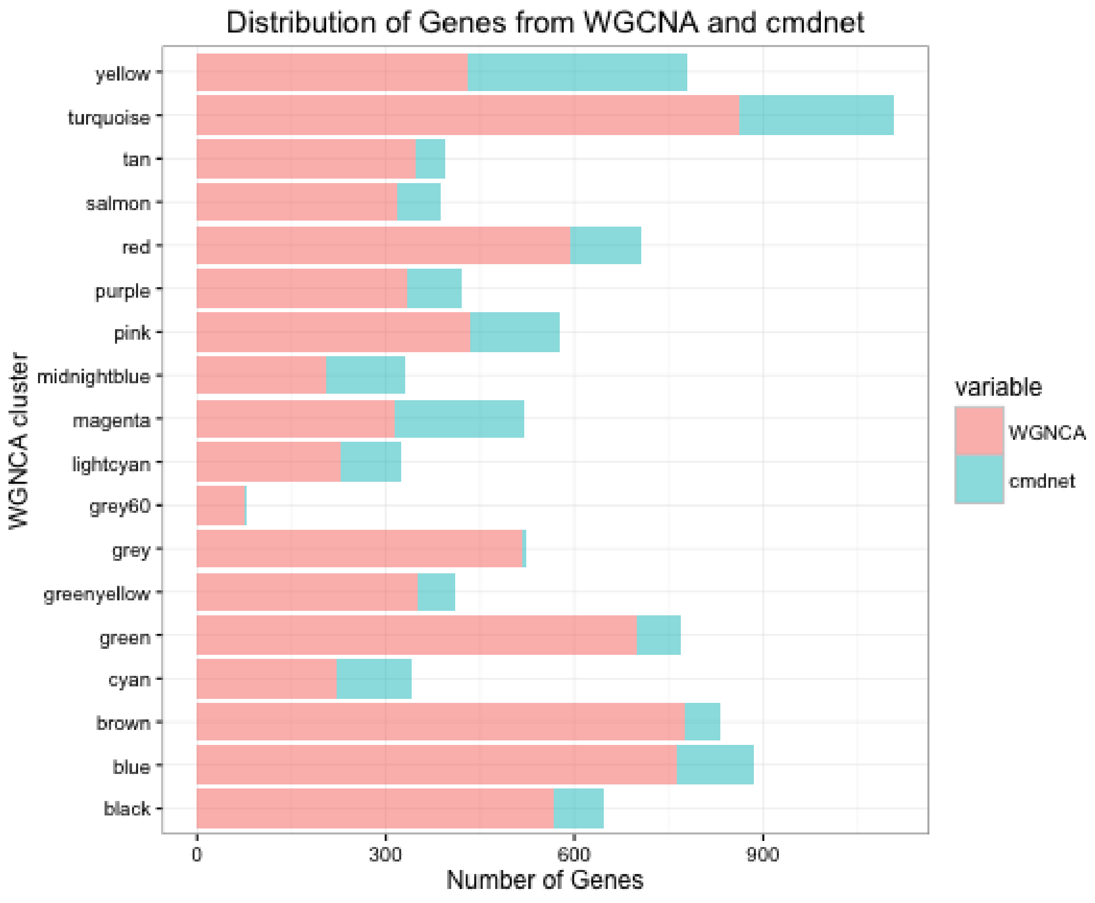


Next, we perform the same task, except this time on the full dataset. We build subnetworks of various sizes using this criteria, and compute the subnetwork’s mean correlation matrix deviation across datasets. To compute the effective noise reduction, we subtract this from the mean correlation matrix deviation found by creating a random subnetwork. We find that maximum noise reduction occurs at around 75 nodes. The half-max is at about 1600 nodes.



Finally, we are interested in knowing whether the least variable genes, the ones selected by `cmdnet`, are clustered in any significant way. For example, it is possible that some of the WGCNA modules have very stable connections, while most of the variability is distributed across other modules. On the other hand, if the most preserved subnetworks include nodes that are evenly distributed across modules, then the de-noising process can be seen as unbiased from the standpoint of WGCNA. This means it would be possible to apply the filter to reduce variability, while still capturing the span of functional groups that make up the systems biology.

To test this hypothesis, we select the top 2000 genes according to the `cmdnet` criteria, and identify which WGCNA modules they belong to. The WGCNA reconstruction here was performed on the average correlation matrix, which was subjected to soft-thresholding before topological overlap was computed. Hierarchical clustering on the topological overlap matrix discovered 18 modules, which were given corresponding color-labels. We then imposed those color-labels on the top 200 genes, which we show in a bar graph. Our result shows that the subnetworks identified by `cmdnet` are evenly distributed across the 18 modules, with the exception of one or two, suggesting that the de-noising is unbiased an maintains system-wide relationships. Interestingly, one of the modules that is deficient in `cmdnet` genes is the grey module, which is reserved for unclustered genes that do not have strong relationships. The other module with few `cmdnet` genes is grey60, which is very small to begin with.

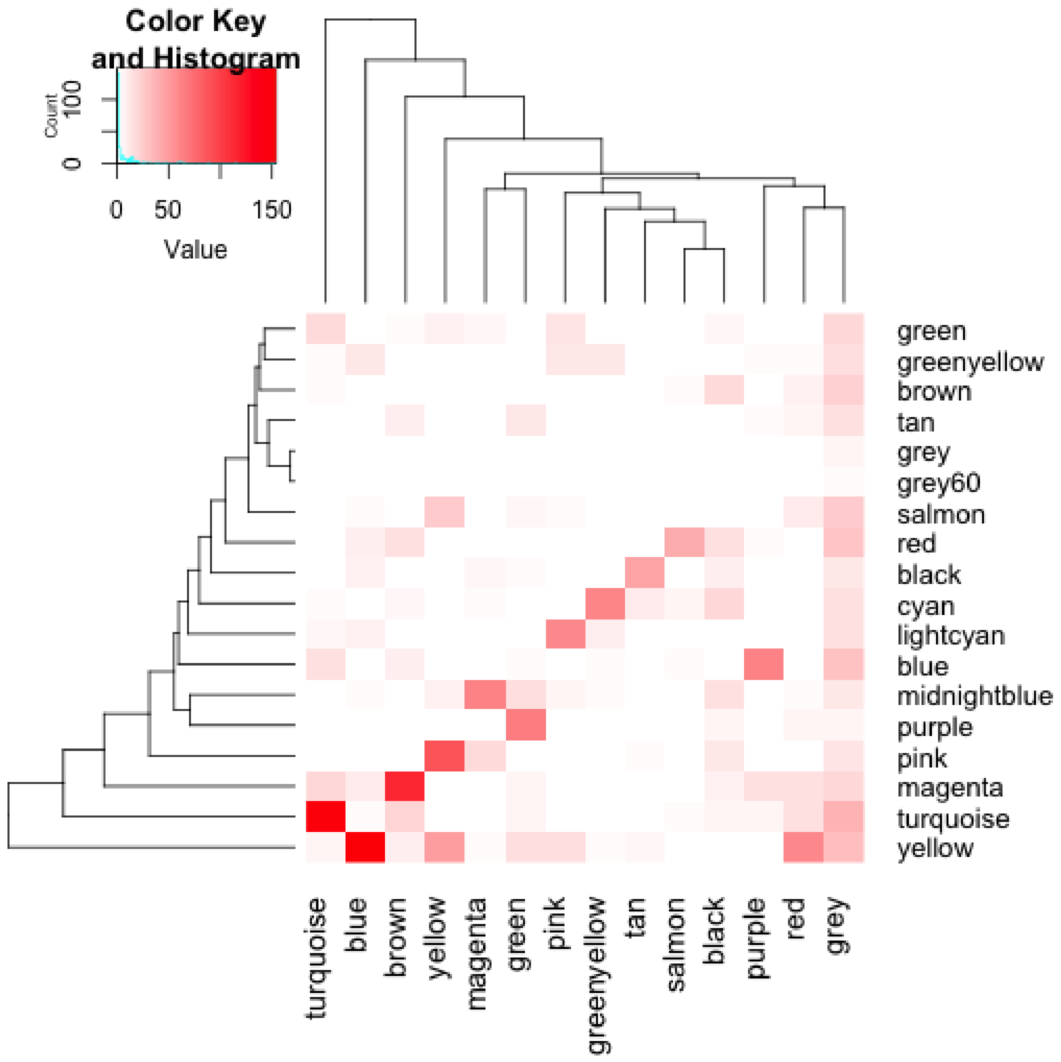


4.2 Pre-Network Reconstruction

We can also use `cmdnet` on the full dataset before any network analysis algorithm is applied. Pre-conditioning filters on variability are already standard practice in genomic analyses. For example, when beginning a whole-transcriptome analysis, an initial array of >40k transcripts is always filtered for transcripts that have low variance, so that unchanging transcripts can be removed before focusing on differential expression.

Just as node expression variability can be used as a filter to improve differential expression analysis, it may be possible to improve network analysis by first filtering on node connectivity variability. When performing network analyses across several datasets, it is common practice to compute networks independently and just hope to find modules that are preserved across networks. Here we propose that `cmdnet` can be used to filter for nodes with the most consistent connectivity, thereby creating correlation matrices that are more similar, which should lead to more reproducible network analyses. Conversely, examining nodes that have high variability in connections may be of interest for identifying differential connectivity across datasets, a notion not yet explored in genomics, but which may prove important.

To examine the feasibility of such a filtering protocol, we apply the `cmdnet` filter prior to network reconstruction by WGCNA. We infer a coexpression network based on the top 2000 genes, which yields 14 modules, and ask how well these modules overlap with the 18 modules defined by full network reconstruction. To visualize the result we use a clustered heatmap, where rows are module labels found by the original full-network reconstruction, and columns are module labels found from the denoised reconstruction. The clustering is able to find a diagonal arrangement, indicating that the module identities are preserved even if denoising is applied before network inference.



5 Discussion

One of the major challenges in genomic research is integration of multiple datasets from the rapidly growing public data repositories. Large variability in protocols, from RNA extraction to data processing, has led to reproducibility concerns. Adding to the variability is an overwhelming number of small sample size experiments, typically no larger than 30 individuals. Of the machine learning methods used to make sense of such high-dimensional, low-sample size datasets, network analysis offers a number of promising tools. Firstly, they act as a means of data compression, identifying a small number of important features through use of clustering and principal component analysis. At the same time, this process provides a kind of noise reduction; since principal components or eigengenes are linear combinations of a large number of variables, they wash out individual fluctuations and keep a more robust composite feature.

Most of this ability comes from leveraging the large number of mutual correlations occuring within the data. But when comparing networks from multiple datasets, identifying stable network features still poses a problem. Understanding the sources of variability in these mutual correlations will lead to a better understanding of both the data, and of the network methods being applied. Here we presented a method for quantifying variability at the network and node-specific levels. By computing correlation matrix distance (CMD) statistics, our `cmdnet` package provides a fast algorithm for the detection of subnetworks preserved across datasets. We found that these subnetworks were evenly distributed across a modular structure inferred by WGCNA, indicating no bias with respect to the network topology. Furtherfore, we found that if the dataset is pre-filtered using `cmdnet` criteria, the resulting modular structure is approximately preserved, suggesting that noise reduction can be applied prior to network reconstruction to improve module preservation across datasets.