How WGCNA can be used to compare and contrast two networks

July 16, 2013

Jeremy Miller
Scientist 1
jeremym@alleninstitute.org
Data and Code Available On-line

Example data and a tutorial describing (in more detail) everything that I will discuss in this section is available from the main WGCNA website: http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/JMiller/

A slightly updated tutorial will be available at the course website or here: http://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/WORKSHOP/

---

Meta-analyses of data from two (or more) microarray data sets.

Jeremy Miller
j Jeremyinla@gmail.com

Microarrays provide expression levels for thousands of genes at once, and therefore have been used extensively to study transcription in the brain. In many cases, the end point for these studies is differential expression analysis: genes A-G are increased and genes H-P are decreased in disease X. Another method for analysis, which is becoming increasingly more common, is gene coexpression analysis: genes Q-S have similar expression patterns. WGCNA is a very useful method for studying gene coexpression, and everything necessary to perform WGCNA successfully can be found at the WGCNA library website: http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/Rpackages/WGCNA/

Given the large number of microarray analyses (sometimes of similar design) one question that may arise is "If group A and group B both ran microarray studies and reported some results, how compatible are these results?" There are currently few methods for comparing results from multiple microarray data sets, but that does not mean that it can't be done. Some methods can be found at the WGCNA website (above). Other comparison methods are listed in this document below, which is a condensed version of the analysis performed in "Miller JA, Horvath S, Geschwind DH. (2010) Divergence of human and mouse brain transcriptome highlights Alzheimer disease pathways. Proc Natl Acad Sci U S A. 2010 Jul 13;107(28):12698-703."

---

Step 1: Getting/loading what you will need for this analysis.

1) Download and install "R" from here: http://cran.r-project.org/
When is it useful to compare networks?

• There are many situations when it is useful to compare two networks:
  • If you have data from two conditions and expect that there will be differences in gene regulation between conditions.
    – control vs. disease
    – cancer state 1 vs. cancer state 2
    – cell type A vs. cell type B
    – frontal cortex vs. occipital cortex
  • If you have microarray data from more than one species, and want to see to what extent gene networks are conserved between species.
  • If you find more than one published data set (i.e., from Gene Expression Omnibus) on comparable tissue and want to find groups of genes that are co-expressed across these data sets.
  • If you have data run on more than one microarray (or RNA-Seq) platform and want to find a platform-independent gene network.

Note that many of the methods that I will describe for comparing two networks can also be used when comparing three or more networks.
Step 1: Data Preparation

Before creating the networks for comparison, R has to be installed and data has to be properly formatted:

• Install R and the required libraries, then start R.
• Collect and import the data from your experiments
  – For this analysis, all of the data is already in an R-object:
    ```r
    load("metaAnalysisData.RData")
    ```

This file contains:
- `datExprA1` and `datExprA2` – two data sets from the Illumina human ref-12 platform
- `datExprB1` and `datExprB2` – data sets from the Illumina human ref-12 and Affymetrix HG-U133A platforms, respectively (datExprA1 and datExprB1 are the same).
- `probesI/A` – probe set IDs for human ref-12 and HG-U133A platforms
- `genesI/A` – gene symbols corresponding to probesI/A

• Preprocess your data
  – For these data, preprocessing has already been done.
• Start the WGCNA library by typing: `library(WGCNA)`
If you are following along with the tutorial...

...you will need to have smaller data sets.

I would recommend going through the tutorial later, but if you prefer to run the tutorial on your laptop as I am going through this section, type:

```r
datExprA1 = datExprA1[1:500,]
datExprA2 = datExprA2[1:500,]
datExprB1 = datExprB1[1:500,]
kpI = intersect(rownames(datExprB2),
               probesA[is.element(genesA,genesI[1:500])])
datExprB2 = datExprB2[kpI,]
```

This *should* make your data set small enough to run real-time.

**Note that you would not normally perform this step, and your results will not look the same as in the tutorial! I’ve run this tutorial with only 500 genes and some of the steps work well, while others don’t work as well.**
Step 2: Ensure that your data are comparable

- For this step, you will find yourself in one of two situations:

  1. Your data sets of interest are from the same platform.
     - If this is the case, congratulations, your data sets are already comparable!
     - You can choose to follow the procedure below, but it is not necessary.

  2. Your data sets of interest are from different platforms.
     - If this is the case, you need to match your probes in some way.
     - The easiest way to do this is to choose one probe for each gene in each data set based on gene symbol using `collapseRows`:

   ```
   datExprB1g = (collapseRows(datExprB1,genesI,probesI))[[1]]
   datExprB2g = (collapseRows(datExprB2,genesA,probesA))[[1]]
   ```

     - *Note that if you are comparing between species, you will have to define genes in species A based on the ortholog from species B.*

- Once you have comparable data, you need to limit your analysis to genes/probes that are expressed in both data sets:

  ```
  commonGenesB = intersect (rownames(datExprB1g),rownames(datExprB2g))
  datExprB1g   = datExprB1g[commonGenesB,]
  datExprB2g   = datExprB2g[commonGenesB,]
  ```

You are now ready to compare networks!
Step 3: Correlating general network properties

A quick way to assess the comparability of two data sets is to correlate measures of average gene expression and overall connectivity between two data sets. The higher the correlations of these properties, the better chance you will have of finding similarities between the two data sets at subsequent stages of the analysis.

Notice three things:
1. The correlations are positive and the p-values are significant in all cases. This suggests that the data sets are comparable.
2. The correlations and p-values are better for expression than for connectivity. This is consistent with many studies.
3. The correlations and p-values for A are better than for B. This is because the two A data sets were run using the same platform. Thus data sets from different platforms are less comparable than data sets from the same platform, but they are still comparable.

If you are following along, replace all of the “5000”s in Step 3 with “50”.
Step 4: Run **WGCNA** on both the data sets

A. Expression values
B. Correlation / Adjacency matrix
C. Topological overlap
   – How close are my neighbors to yours?
D. Hierarchical clustering to group genes into modules
E. Determine hub genes from intramodular connectivity
Step 4: Run WGCNA on both the data sets

- The next several steps of the analysis are the same for data sets run on the same platform as those run on different platforms.
  - This is because we made our data comparable in step 2!
  - Since the results are more significant for the within-platform comparisons, I will focus on data sets A1 and A2.

- We performed WGCNA on both networks, and:
  - Both the networks look good
  - For this analysis we only assigned modules in a reference network (A1).
    - If you have a “control” data set, use this one as your reference!

Note that for comparing networks it is usually (but not always) best to have a small number of large modules.
Step 4½: Calculate and visualize principal components

Module eigengenes for the reference network are required for quantitative assessments later in this analysis:

Contains ALL principal components

Module eigengene step

(The other variables in this code are for visualizations)

PCs1A = moduleEigengenes(t(datExprAlg), colors=modulesA1)
ME_1A = PCs1A$eigengenes
distPC1A = 1-1-(1+cor(ME_1A,use="p"))/2
distPC1A = ifelse(is.na(distPC1A), 0, distPC1A)
pcTree1A = hclust(as.dist(distPC1A),method="a")
MDS_1A = cmdscale(as.dist(distPC1A),2)
colorsA1 = names(table(modulesA1))

Note: Visualizations will be discussed at another time, and are not critical for understanding between-network analyses.
Step 5: Assess module preservation

- Module preservation can be done in a quantitative way using \texttt{modulePreservation} (as discussed in an earlier lecture).
- This can also be done in a qualitative way (particularly if the data sets in your analysis have very similar network structure):

\begin{verbatim}
plotDendroAndColors(geneTreeA1, modulesA1, "Modules", dendroLabels=FALSE, hang=0.03, addGuide=TRUE, guideHang=0.05, main="Gene dendrogram and module colors (A1)")
plotDendroAndColors(geneTreeA2, modulesA1, "Modules", dendroLabels=FALSE, hang=0.03, addGuide=TRUE, guideHang=0.05, main="Gene dendrogram and module colors (A2)")
\end{verbatim}

- Note that we impose the module definitions from A1 onto the A2 network

Since these module labels still group together in A2, there is generally very good preservation.
Step 6: \( k_{ME} \) and its use in comparing networks

Module membership (\( k_{ME} \)) is useful, in that it can be used to measure correlations between each gene and each ME, and thus even genes which were not initially assigned to a module can be included in between-network comparisons.

Step 6a: Obtain the module membership values

- First find the \( k_{ME} \) values in the reference network
- Then find the \( k_{ME} \) values in the other network using the module assignments from the reference network.

```r
geneModuleMembership1 = signedKME(t(datExprAlg), ME_1A)
colnames(geneModuleMembership1)=paste("PC",colorsAl,".cor",sep="")

MMPvalue1 = corPvalueStudent(as.matrix(geneModuleMembership1),
dim(datExprAlg)[[2]]);
colnames(MMPvalue1)=paste("PC",colorsAl,".pval",sep="");

Gene = rownames(datExprAlg)
kMEtable1 = cbind(Gene,Gene,modulesA1)
for (i in 1:length(colorsA1)) kMEtable1 = cbind(kMEtable1, geneModuleMembership1[,i],
MMPvalue1[,i])
colnames(kMEtable1)=c("PSID","Gene","Module",sort(c(colnames(geneModuleMembership1), colnames(MMPvalue1)))))

# REPEAT THIS CODE FOR NETWORK A2 USING MODULES FROM A1```

Get \( k_{ME} \) values

Get P-values

Output in correct format for other functions.
Step 6: $k_{ME}$ and its use in comparing networks

Step 6b: Plot the module membership values between networks

- In general, modules showing high $k_{ME}$ correlation are highly preserved
- These values can be plotted in two similar (but not identical) ways:

(Note that these plots would be available for every module in the reference network.)

Using all genes (left) allows one to include all positively and negatively correlated genes, but often also includes a lot of noise (although not in this case). Using only genes in the module (right) is a visual way of assessing hub gene conservation: if these genes show between-set correlation, then the genes in the upper right of the plot are likely to be common hub genes between data sets.
Step 6: $k_{ME}$ and its use in comparing networks

Step 6c: Finding hub genes common to both networks

- Common hub genes are often very useful
  - For example, one could find genes critical to energy metabolism pathways in both mouse and human, or key oncogenesis genes in multiple cancer types.
- It is often just as important finding genes unique to a single network.
  - This can also be done using $k_{ME}$, for example by taking the difference in ranked $k_{ME}$ instead of the maximum.

```r
Gene = rownames(datExprA1g)
topGenesKME = NULL
for (c in 1:length(colorsA1)){
kMErank1 = rank(-geneModuleMembership1[,c])
kMErank2 = rank(-geneModuleMembership2[,c])
maxKMErank = rank(apply(cbind(kMErank1,kMErank2+.00001),1,max))
topGenesKME = cbind(topGenesKME,Gene[maxKMErank<=10])
}
colnames(topGenesKME) = colorsA1
topGenesKME
```

Ranked $k_{ME}$ is used to avoid issues of normalization between data sets.

This code will output the top 10 hub genes for each module to the screen, but in principle, a sorted list of any number of top genes could be outputted to a file.

Example output...

```
   block        blue      brown       green      grey       magenta
[1,] "16-Sep"  "ACTR10"  "CADPS"    "CYBRD1"   "CAMKIV"   "AHSA1"
[2,] "Cclrf19"  " ATP5O"  " GABBR2"  " DPEYSL3"  " CYFIP2"   "DNAJA1"
[3,] "DA2AF2"   " COX6B1"  " MAP7D2"  " FERM2T"   " DLG2"     "DNAJB1"
[4,] "FN1A"     " KIAA1279" " NABB"     " GRAMD3"   " DNM1"     "DNAJB6"
[5,] "LIT1AF"   " LOC666630" " NECAP1"   " MECAM"    " HSPA1A2"   "FKBP4"
[6,] "LOC145815" " PRDX3"    " NSF"      " IL17D"     " MAP1A"    "HSPA1A"
[7,] "LOC440526" " TBCA"     " F0M2L1"   " LOC219854" " HCN2"     "HSPA1B"
[8,] "LOC730536" " TMEH85"   " PRKPL"    " MAPRE1"    " PGK1"      "HSF1"
[9,] "STIP2D1"  " TXN"      " SLC39A6"  " NIK2"      " SCAMP5"    "HSPO1"
[10,] "SGCE"    " NRBB"     " ND7"      " RHOBTB3"   " Uncsa"     "STIP1"
```
Step 7: Network annotation and comparison

Methods for network and module annotation will be discussed in detail in a different lecture. I will focus this discussion on how to use these annotations to compare networks.

Step 7a: Module comparisons using VisANT (similarities)
VisANT (and Cytoscape) can help find common hubs in a module between two networks, as well to find hubs specific to a single network.

- While VisANT makes great comparison plots, this method only works a small percentage of the time.

In this case, we ran VisANT using the same genes in both networks and DNM1 pops out as a hub in both cases.
Step 7: Network annotation and comparison

Step 7a: Module comparisons using VisANT (differences)

VisANT (and Cytoscape) can also be used find modules unique to a single network (in this case, A1). The basic method is as follows:

1. Make the VisANT plot using TO in A1 (as before)
2. Find the TO ratios between networks. This is basically a scaled measure of difference in connection strength between two genes.
3. Remake the VisANT plot, this time omitting all connections that do not have a very high TO ratio.

In this case, SCAMP5 is specific to the pink module in network A1 since most of its connections have high TO ratio and are thus differential connections.
Step 7: Network annotation and comparison

Step 7b: Comparing networks based on gene list enrichment

- For this analysis, gene lists are defined only in network A1, and thus the resulting enrichments will be identical for both networks.
- Modules can (and should!) still be annotated using the various methods (described in a different lecture):
  - DAVID / EASE
  - userListEnrichment / GOenrichmentAnalysis
  - Ingenuity
  - Chilibot, WebGestalt, UGET
  - Galaxy
  - ToppGene
  - etc.

- We will briefly revisit this section later when discussing networks with different module definitions.
Step 8: Using phenotypic traits to compare networks

This step can be broken down into three parts:
1. Collect phenotypic traits common to both networks
2. Perform within-network analyses and visualize
3. Perform between-network comparisons and visualize

Step 8a: Collect phenotypic traits common to both networks
Here are some examples of situations when this would be useful:
1. You have one mouse and one human brain data set.
   – Use information about brain region to find genes differentially expressed between compartments of hippocampus in both species.
   – Also, look for genes showing different patterns in mouse and human.
2. You have cancer data sets from several labs.
   – Collect information on prognosis to find genes that consistently show increased expression in survival cases.
   – Also, look for genes that correlate with prognosis only in certain types of cancers.
Step 8: Using phenotypic traits to compare networks

Step 8b: Perform within-network analyses and visualize

**Traits:**
Age, gender, state, disease vs. control, tissue type, etc.

**Same traits:**
Age, gender, state, disease vs. control, tissue type, etc.

**Relate traits to gene expression:**
correlate eigengene with age, etc.

**Relate traits to gene expression:**
correlate eigengene with age, etc.

**Compare:**
Are there similar relationships in both data sets?
Step 8: Using phenotypic traits to compare networks

Step 8b: Perform within-network analyses and visualize (example 1)

Here are two gene-centric examples in the case of network A1

*Identical analyses can be done use eigengenes instead of genes*

```r
# Find the top 3 region genes
var    = list(region=="CA1", region=="CA3")
datReg = t(apply(datExprAlg,1,t.test.l))
colnames(datReg)=c("MeanCA1","MeanCA3","SD_CA1","SD_CA3", "PvalRegion")
datReg[order(datReg[,5])[1:3],]
  #          MeanCA1   MeanCA3   SD_CA1  SD_CA3  PvalRegion
  # NRIP3  10509.8842  22481.7087 2951.5061 6239.7370 1.335158e-06
  # KCNH3   3814.7772  1840.0098 1049.7946  749.2281 1.493591e-06

# Find the top 3 age genes
var    = age
datAge = t(apply(datExprAlg,1,cor.test.l))
colnames(datAge)=c("CorrAge","PvalAge")
datAge[order(datAge[,2])[1:3],]
  #          CorrAge      PvalAge
  # DDX42     0.7116889 4.947339e-06
  # SCD       0.6961764 9.660017e-06
  # KIAA0907  -0.6818583 1.728760e-05
```

We can then compare these lists and plots with our other data set!
Step 8: Using phenotypic traits to compare networks

Step 8b: Perform within-network analyses and visualize (example 2)

• Another way of doing this analysis is to directly compare lists of differentially expressed genes between data sets.

• For example, one can find between-species and species-specific markers for brain regions using the following strategy:
  – Find differential expression across brain regions in mouse and human
  – Take the subset of these genes with large fold changes (up or down).
  – For each brain region, take the genes in both mouse and human lists
    • Genes up in both species are common markers
    • Genes up in human and down in mouse are human-specific markers
    • Genes up in mouse and down in human are mouse-specific markers

<table>
<thead>
<tr>
<th>Gene</th>
<th>Region</th>
<th>FC_mouse</th>
<th>FC_Human1</th>
<th>FC_Human2</th>
<th>Score</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSPB3</td>
<td>Intralaminar nuclei of the dorsal thalamus</td>
<td>6.33853</td>
<td>0.25876</td>
<td>0.10329</td>
<td>3.86</td>
<td>MouseOnly</td>
</tr>
<tr>
<td>HTR4</td>
<td>Caudoputamen</td>
<td>0.26748</td>
<td>3.25819</td>
<td>3.09481</td>
<td>3.09</td>
<td>HumanOnly</td>
</tr>
<tr>
<td>IGSF1</td>
<td>Pallidum_dorsal region</td>
<td>0.15454</td>
<td>4.59034</td>
<td>3.31997</td>
<td>3.32</td>
<td>HumanOnly</td>
</tr>
<tr>
<td>IRX2</td>
<td>Pallidum_dorsal region</td>
<td>0.02373</td>
<td>30.00446</td>
<td>15.23243</td>
<td>15.23</td>
<td>MouseOnly</td>
</tr>
<tr>
<td>ITGA2</td>
<td>Pallidum_dorsal region</td>
<td>0.05533</td>
<td>4.37464</td>
<td>3.99156</td>
<td>3.99</td>
<td>HumanOnly</td>
</tr>
<tr>
<td>KCNAB1</td>
<td>Caudoputamen</td>
<td>3.33912</td>
<td>6.38032</td>
<td>9.49088</td>
<td>3.34</td>
<td>Human&amp;Mouse</td>
</tr>
</tbody>
</table>

Here, we expect to see IRX2 highly expressed in the dorsal pallidum (aka globus pallidus) in human, but not in mouse.

*Can we confirm this using Allen Institute data?*
IRX2 does show species-specific expression

Both probes for IRX2 show high expression in human globus pallidus:

In mouse, there is no expression of IRX2 in globus pallidus.
• This probe is expressed elsewhere in the brain, so we know it works.
Step 8c: Perform between-network comparisons and visualize

We can also directly compare matched traits between data sets, for example by plotting expression levels of genes or module eigengenes. In this case, we plot data from matched brain regions across two adult human brains.

In this example, both ADCYAP1 and PSD2 have expression levels in dorsal thalamus that are 2-4 fold higher than hippocampus in brain 1 (x-axis). With the addition of brain 2, we see that only the differential expression found in ADCYAP1 is consistent between brains.

Thus this method allows us to visualize gene expression across all traits and find genes with consistent expression between networks in a single graph!
Comparing networks with different module definitions

Thus far in the meta-analysis we have compared networks in which the module identifiers for genes from one network are assigned to be module identifiers for the same genes in a second network.

Another common occurrence is to have two networks with different sets of module identifiers.
When to define modules differently in networks

There are advantages and disadvantages for defining modules separately in a second network, and the best method should be decided on a case-by-case basis.

• Advantages of using same module definitions between networks:
  – It is easier to keep track of one set of module definitions than two.
  – If you have a reference, everything can be easily compared against it.

• Advantages of having distinct module definitions in each network:
  – You can find modules unique to any network
  – Hub genes, annotations, etc. can be found for all networks independently.

What additional network comparisons can we make with a second set of modules?
Assess module overlap between networks

Once we have our two networks, we can check to see which modules in network A1 contain a significant number of overlapping genes with modules in B2 using `matchLabels`. 
- This function reassigns model labels in network B2 such that corresponding modules are assigned the same color.
- For example, the yellow modules below have a significant number of overlapping genes, ⭐ while the purple module in B2 is specific to B2. ▲
- The specific genes common to each module, as well as the p-value of overlap can then be found using `userListEnrichment`. (Also see `overlapTable`.)
Comparing module annotations

- Comparing module annotations is another way to determine module agreement between networks.
- This can be done by performing nearly any annotation strategy (GO enrichment, userListEnrichment, etc.), and then looking to see which modules have shared annotation.
  - For example, if the green module in A1 and the purple module in B2 both had an over-representation for mitochondrial genes, one may want to consider these modules as related, even if they don't share a significant number of common genes.
  - Likewise, if the yellow modules in A1 and B2 both are enriched for astrocyte genes, that provides further evidence that these genes have common function.

Note: there is a whole set of tutorials on consensus networks at the WGCNA website (which I am not going to discuss here):
Summary of between-network analysis

**Step 1:** Data Preparation
**Step 2:** Ensure that your data are comparable
**Step 3:** Correlating general network properties
**Step 4:** Run WGCNA on both the data sets
**Step 5:** Assess module preservation
**Step 6:** kME and its use in comparing networks
  - Obtain the kME values
  - Plot the module membership values between networks
  - Finding hub genes common to both networks
**Step 7:** Network annotation and comparison
  - Module comparisons using VisANT
  - Comparing networks based on gene list enrichment
**Step 8:** Using phenotypic traits to compare networks
  - Collect phenotypic traits common to both networks
  - Perform within-network analyses and visualize
  - Perform between-network comparisons and visualize

**Comparing networks with different module definitions**
1. Assess module overlap between networks
2. Comparing module annotations
Acknowledgements

We wish to thank the Allen Institute founders, Paul G. Allen and Jody Allen, for their vision, encouragement, and support.

Any questions?

Steve Horvath
Dan Geschwind
Mike Hawrylycz
Peter Langfelder
Mike Oldham