Supplementary Materials and Methods for:

Michael C. Oldham\textsuperscript{1,4}, Steve Horvath\textsuperscript{*2,3}, and Daniel H. Geschwind\textsuperscript{*3,4} (2006).

Conservation and Evolution of Gene Co-expression Networks in Human and Chimpanzee Brains.

\textsuperscript{1}Interdepartmental Program for Neuroscience, \textsuperscript{2}Department of Biostatistics, \textsuperscript{3}Department of Human Genetics, and \textsuperscript{4}Department of Neurology, Program in Neurogenetics, and The Neuropsychiatric Institute, UCLA School of Medicine, 710 Westwood Plaza, Los Angeles, CA 90095-6814, USA

Email addresses: oldham@ucla.edu, shorvath@mednet.ucla.edu, dhg@ucla.edu

Statistical correspondence: shorvath@mednet.ucla.edu
Summary

We provide additional Materials and Methods as well as the statistical software code used for generating the weighted gene co-expression network results. Thus, the reader should be able to reproduce all of our findings. This document also serves as a tutorial for weighted gene co-expression network analysis. Some familiarity with the R software (http://www.r-project.org/) is desirable, but the document is fairly self-contained. This document along with customized R functions and the accompanying data files can be found at the following web page: http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/HumanChimp. More material on weighted network analysis can be found here: http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/.
Analysis of microarray data

The dataset used for network construction consisted of 36 Affymetrix HGU95Av2 microarrays surveying gene expression with 12,625 probe sets in three adult humans and three adult chimpanzees across six matched brain regions: Broca's area, anterior cingulate cortex, primary visual cortex, prefrontal cortex, caudate nucleus, and cerebellar vermis (1). Because the arrays used in this study were designed for human mRNA sequences, it is necessary to account for the effect of interspecies sequence differences on chimpanzee gene-expression values (2). All probes on the array were compared against the human genome (Build 34) and the chimpanzee draft genome using MegaBlast (http://www.ncbi.nlm.nih.gov/BLAST/). Any probe without a perfect match in both species (approximately 1/4) was masked during the calculation of expression values (GCOSv1.2, Affymetrix, Inc). In addition, only probe sets with six or more matching probes were retained for subsequent analyses (n=11,768/12,625). For each array, expression values were scaled to an average intensity of 200 (GCOSv1.2, Affymetrix, Inc.). Quantile normalization was then performed and inter-array correlations were calculated using R. Following quantile normalization, the average inter-array correlation was 0.924 among all 18 human arrays and 0.937 among all 18 chimpanzee arrays. In specific instances in which the composition of this dataset was altered, the removal of a brain region(s) took place prior to quantile normalization.

Choice of genes for network analysis

For computational reasons, network analysis was limited to 4000 probe sets. (Note: although some genes are represented by multiple probe sets and other probe sets are not fully annotated, for consistency we refer to probe sets as "genes" throughout the manuscript, unless otherwise
noted.) In order to enrich this subset with genes likely to play important roles in the brain, a non-neural "filter" was applied to identify genes with greater variance in neural versus non-neural tissue. A publicly available microarray dataset was obtained for human lung (3). The human lung dataset consisted of 18 Affymetrix HGU95Av2 microarrays measuring gene expression in normal adult human lung tissue, and was normalized as described above. The same probe mask file that was used for the human and chimpanzee brain datasets was also applied to the human lung dataset. After scaling all arrays to the same average intensity (200), the mean inter-array correlation for one human lung array (CL2001032718AA) was 2.91 SD below the average and was removed from the dataset. Following quantile normalization in R, the average inter-array correlation for the human lung dataset was 0.925. For each probe set on the microarray, the variance was calculated in the non-neural (lung) and neural (brain) datasets for the human samples. All probe sets were ranked according to their variance in both the neural and non-neural datasets, and each probe set's rank in the neural dataset was subtracted from its rank in the non-neural dataset. These rank differences (sorted from high to low) were used to select the top 4000 probe sets among all probe sets showing greater variance in human brain than human lung. In cases where <4000 probe sets showed greater variance in brain than lung, the top 4000 probe sets were selected solely on the basis of their rank differences between the neural and non-neural datasets.
Overview of methodology for constructing gene co-expression networks

For a graphical overview of the methodology described here, see Figure S1. Network analysis of gene-expression data begins with the understanding that the information captured by microarray experiments is far richer than a list of differentially expressed genes. Rather, microarray data are more completely represented by considering the relationships between measured transcripts, which can be assessed by simple correlations (Figure S1A). In most microarray data analyses, however, these relationships go essentially unexplored. We set out to construct and compare gene co-expression networks in normal adult human and chimpanzee brains using microarray data. The dataset used for network construction consisted of matched human and chimpanzee samples from six brain regions: Broca's area, anterior cingulate cortex, primary visual cortex, prefrontal cortex, caudate nucleus, and cerebellar vermis (1). To construct the gene co-expression networks, we calculated all possible pairwise Pearson correlations for 4000 genes in human and chimpanzee brains in parallel. Because microarray data can be noisy and the number of samples is often small, correlations were weighted by raising their absolute value to a power, $\beta$ (4) (Figure S1B). A weighted correlation between two genes, in turn, represents the connection strength between those genes in the network. Summing the connection strengths for each gene with all other genes resulted in a single number (called network connectivity, or $k$) that represents how strongly that gene is connected to all other genes in the network (Figure S1B). We constructed networks in humans and chimpanzees and compared $k$ between the species to evaluate network conservation and its dependence on specific brain regions.

A major goal of network analysis is to identify groups, or "modules", of densely interconnected genes (5). Such groups are often identified by searching for genes with similar patterns of
connection strengths to other genes, or high "topological overlap" (4, 6) (Figure S1C). It is important to recognize that correlation and topological overlap are very different ways of describing the relationship between a pair of genes: while correlation considers each pair of genes in isolation, topological overlap considers each pair of genes in relation to all other genes in the network. More specifically, genes are said to have high topological overlap if they are both strongly connected to the same group of genes in the network (i.e. they share the same "neighborhood"). Topological overlap thus serves as a crucial filter to exclude spurious or isolated connections during network construction (see asterisk in Figure S1C; for a comparison of networks constructed by correlation and topological overlap, see Figure S2). To calculate the topological overlap for a pair of genes, their connection strengths with all other genes in the network are compared. By calculating the topological overlap for all pairs of genes in the network, modules can be identified (Figure S1D). In the resulting dendrogram, discrete branches of the tree correspond to modules of co-expressed genes. The network can also be represented by a standard multi-dimensional scaling plot (Figure S1E), in which each point represents a gene and increasing proximity reflects increasing topological overlap. Modules can be explored within the context of the global network, or individually as smaller, local networks, in which case the steps described in (B) and (C) are repeated while only considering genes within a given module. In this case, intramodular connectivity, or $k_{in}$, is calculated as the sum of a gene's connection strengths with all other genes in its module, providing a basis for visualizing module structure (Figure S1F). Using this approach, we identified a number of modules, each corresponding to distinct regions of the human brain. We assessed the overall conservation of each module in chimpanzees and identified specific genes with vastly different positions and connections in human and chimpanzee brain networks. By comparing the topological overlap of
genes within each module between humans and chimpanzees, we were able to identify gene co-expression relationships that are present in the human brain but absent in the chimpanzee. Through comparisons of human and chimpanzee DNA sequence, we relate these differences to evolutionary changes at the genomic level.

**Weighted gene co-expression network construction in human and chimpanzee brains**

Weighted network construction was performed using R (http://www.r-project.org/) as described (4). For all comparisons the human and chimpanzee sample sizes were equal. The absolute values of the Pearson correlation coefficients were calculated for all pairwise comparisons of gene-expression values across all human samples and all chimpanzee samples. The correlation matrix for each species was then transformed into a matrix of connection strengths (i.e. an "adjacency" matrix) using a power function (connection strength=|correlation|^β), which resulted in a "weighted" network (4). The use of weighted networks represents an improvement over unweighted networks produced by dichotomizing the Pearson correlation matrix, since a) the continuous nature of the gene co-expression information is preserved, and b) the results of weighted network analyses are highly robust with respect to the choice of the parameter β, whereas unweighted networks display sensitivity to the choice of the cutoff (4). Zhang and Horvath (4) proposed a scale-free topology criterion for choosing β, which was applied here. In order to make meaningful comparisons across datasets, a power of β=9 was chosen for all analyses. By comparison, this choice of β is similar to constructing an unweighted network that dichotomizes the correlation matrix with a threshold of 0.90. Using the Fisher transformation of the correlation matrix and the sample size in this dataset, one can show that this threshold is highly significant (p<5.0E-22), providing a strong foundation for this analysis. We note,
however, that our results were highly robust with respect to the choice of $\beta$ (data not shown).

The overall connectivity for each gene ($k$) is the sum of the connection strengths ($|\text{correlation}|^\beta$) between that gene and all other 3999 genes in the network, scaled to lie between 0 and 1. The intramodular connectivity for each gene ($k_{in}$) is the sum of the connection strengths between that gene and all other genes in its module, scaled to lie between 0 and 1 (see module description below). Unless otherwise noted, all correlations described in Results were performed using Spearman's rank correlation.

Detection and characterization of modules

Module detection and characterization were performed using customized R software functions (see below). Because modules consist of highly connected genes (7) and their identification is computationally intensive, we further restricted our analysis by retaining only those genes for which $k$ was $> 0.1$ in at least one species ($n=2241$). Gene co-expression modules in the human brain were identified using average linkage hierarchical clustering to group genes based upon 1 - the topological overlap of their network connection strengths (4, 6). Modules roughly correspond to branches of the dendrogram as visualized in the topological overlap matrix (TOM) plot, which orders genes according to their position in the dendrogram and displays the degree of topological overlap on a colored scale, with increasing color intensity corresponding to increasing similarity. In the TOM plot, modules correspond to "squares" along the diagonal (see p. 21 of this document for an example). Modules were also visualized by classical multi-dimensional scaling in three dimensions. For each module, a "heat map" was produced with rows corresponding to genes and columns corresponding to samples (ordered by brain region; red indicates increased expression, black neutral expression, green decreased expression). The
gene-expression profile of each module was decomposed via singular value decomposition 
($X = UDV^T$) and the value of the first module eigengene, $V_1$, was plotted for each sample. The 
first module eigengene explained 56% of the variance in the turquoise module, 47% in the blue 
module, 53% in the brown module, 63% in the yellow module, 45% in the green module, 62% in 
the red module, 68% in the black module, and 58% in the visual cortex module. In order to test 
the significance of module characterizations, an indicator variable was defined for each brain 
region. For example, in the turquoise module, 1 corresponded to cerebellar samples and 0 to 
non-cerebellar samples. We then compared the average redness (i.e. first module eigengene) in 
the cerebellar samples versus the non-cerebellar samples using the Kruskal-Wallis test.
References


Getting the software and data

1. Downloading the R software: go to http://www.r-project.org/, download R and install it on your computer. After installing R, you will need to install several additional R library packages. For example, to install the package "Hmisc", open R, go to the "Packages" menu and select "Install packages...", then choose Hmisc. R will automatically install the package. Do the same for the other libraries mentioned below. Note that several libraries are already present in the software so there is no need to re-install them.

2. Download the zip file “Zipped datasets.zip” at http://www.genetics.ucla.edu/labs/horvath/CoexpressionNetwork/HumanChimp/ and unzip it into a directory. Datasets 1-4 are derived from reference (1). Dataset 5 is derived from references (8, 9). Descriptions below:

- Dataset 1: Primary dataset used for network construction. Consists of six matched brain regions between humans and chimpanzees (four cortical, caudate nucleus, and cerebellum).
- Dataset 2: Consists of five matched brain regions between humans and chimpanzees (four cortical and cerebellum)
- Dataset 3: Consists of five matched brain regions between humans and chimpanzees (four cortical and caudate nucleus)
- Dataset 4: Consists of four matched brain regions between humans and chimpanzees (four cortical)
- Dataset 5: Dataset used for network validation. Consists of a variety of partially matched cortical brain regions between humans and chimpanzees.


4. Unzip all the files into the same directory (e.g. we put it into “C:/Documents and Settings/HumanChimpNetwork”).
# STARTING THE R session:
# Open the R software by double-clicking the desktop icon.
# To interact with the R software, copy and paste the commands into the R console.
# Text after "#" is a comment and is automatically ignored by R.

# Set the working directory of the R session by using the following command:
setwd("C:/Documents and Settings/HumanChimpNetwork")
# Note that we use / instead of \ in the path.

# Read in the R libraries:
library(MASS)  # standard, no need to install
library(class) # standard, no need to install
library(cluster)
library(sma)   # install it for the function plot.mat
library(impute) # install it for imputing missing value
library(Hmisc) # probably you won’t need this
library(splines) # probably, you won’t need this

# Memory
# Check the maximum memory that can be allocated:
memory.size(TRUE)/1024
# Increase the available memory:
memory.limit(size=4000)

# Read in the custom network functions:
source("NetworkFunctions.txt")

## Read in array data:
dat1=read.csv("Dataset 1 (network construction).csv",header=T)
attach(dat1)
dim(dat1)
# This data frame contains the gene expression data.
# By our convention, columns are genes and rows are samples.
datExpr=data.frame(t(dat1[Brain_variant_H>0,2:39]))
dim(datExpr)
dimnames(datExpr)[[1]]
indexHuman=c(19:36)
indexChimp=c(1:18)
## Choice of power for weighting the Pearson correlation matrix

To choose a power beta for computing the connection strengths, we make use of the
the Scale-free Topology Criterion (Zhang and Horvath 2005). We focus on the scale-free
 topology model fitting index (denoted as scale.law.R.2) that quantifies how well
a network satisfies a scale-free topology.

The slope of the regression corresponds to the value gamma for the scale free distribution.

```r
powers1=c(seq(1,10,by=1),seq(12,20,by=2))
RpowerTable=PickSoftThreshold(datExpr[indexHuman,], powervector=powers1,
RsquaredCut=0.90)[[2]]
```

### Output

<table>
<thead>
<tr>
<th>Power</th>
<th>scale.law.R.2</th>
<th>slope truncated.R.2 mean.k. median.k. max.k.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6690</td>
<td>2.470</td>
</tr>
<tr>
<td>2</td>
<td>0.0267</td>
<td>0.360</td>
</tr>
<tr>
<td>3</td>
<td>0.1190</td>
<td>-0.380</td>
</tr>
<tr>
<td>4</td>
<td>0.5460</td>
<td>-0.784</td>
</tr>
<tr>
<td>5</td>
<td>0.7260</td>
<td>-1.020</td>
</tr>
<tr>
<td>6</td>
<td>0.8130</td>
<td>-1.170</td>
</tr>
<tr>
<td>7</td>
<td>0.8640</td>
<td>-1.270</td>
</tr>
<tr>
<td>8</td>
<td>0.8880</td>
<td>-1.340</td>
</tr>
<tr>
<td>9</td>
<td>0.9150</td>
<td>-1.370</td>
</tr>
<tr>
<td>10</td>
<td>0.9190</td>
<td>-1.410</td>
</tr>
<tr>
<td>11</td>
<td>0.9420</td>
<td>-1.440</td>
</tr>
<tr>
<td>12</td>
<td>0.9420</td>
<td>-1.460</td>
</tr>
<tr>
<td>13</td>
<td>0.9490</td>
<td>-1.460</td>
</tr>
<tr>
<td>14</td>
<td>0.9480</td>
<td>-1.450</td>
</tr>
<tr>
<td>15</td>
<td>0.9560</td>
<td>-1.450</td>
</tr>
</tbody>
</table>

# To visualize how the scale-free fit and the mean connectivity depend on
# the power parameter beta, we make use of the following code:

```r
collect_garbage()
cex1=0.7
par(mfrow=c(1,2))
plot(RpowerTable[,1], -sign(RpowerTable[,3])*RpowerTable[,2],xlab="Soft Threshold (power)",ylab="Scale Free Topology Model Fit, signed R^2",type="n")
text(RpowerTable[,1], -sign(RpowerTable[,3])*RpowerTable[,2], labels=powers1,cex=cex1,col="red")
abline(h=0.9,col="red")
plot(RpowerTable[,1], RpowerTable[,5],xlab="Soft Threshold (power)",ylab="Mean Connectivity", type="n")
text(RpowerTable[,1], RpowerTable[,5], labels=powers1, cex=cex1,col="red")
```
To choose a cutoff value \( \tau \), we make use of the Scale-free Topology Criterion (Zhang and Horvath 2005). The criterion focuses on the linear regression model fitting index \((\text{denoted as scale.law.R.2})\) that quantifies how well a network satisfies a scale-free topology. Consider the left-hand side: we choose the smallest power where the \( R^2 \) curve seems to saturate. The horizontal red line corresponds to \( R^2 = 0.90 \). From the above table, we find that the resulting slope (minus the gamma parameter of the scale-free plot) looks OK. We and others have found that the slope should lie between -1 and -2. It is worth emphasizing that our findings are highly robust with respect to the power. To check this, consider different values for \( \text{powerHuman} \) below:

Here the scale-free topology criterion with a \( R^2 \) threshold of 0.90 leads us to pick a power of 9.

\[ \text{powerHuman}=9 \]
# For completeness, we also report the scale-free fits for the chimp network:
RpowerTableChimp=PickSoftThreshold(datExpr[indexChimp,], powervector=powers1, RsquaredCut=0.90)[[2]]

Output:

<table>
<thead>
<tr>
<th>Power scale.law.R.2</th>
<th>slope truncated.R.2</th>
<th>mean.k.</th>
<th>median.k.</th>
<th>max.k.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.163</td>
<td>0.973</td>
<td>0.946</td>
<td>1150.00</td>
</tr>
<tr>
<td>2</td>
<td>0.123</td>
<td>-0.432</td>
<td>0.952</td>
<td>502.00</td>
</tr>
<tr>
<td>3</td>
<td>0.698</td>
<td>-0.902</td>
<td>0.975</td>
<td>266.00</td>
</tr>
<tr>
<td>4</td>
<td>0.828</td>
<td>-1.100</td>
<td>0.976</td>
<td>160.00</td>
</tr>
<tr>
<td>5</td>
<td>0.885</td>
<td>-1.210</td>
<td>0.980</td>
<td>104.00</td>
</tr>
<tr>
<td>6</td>
<td>0.909</td>
<td>-1.300</td>
<td>0.979</td>
<td>71.50</td>
</tr>
<tr>
<td>7</td>
<td>0.931</td>
<td>-1.330</td>
<td>0.984</td>
<td>51.60</td>
</tr>
<tr>
<td>8</td>
<td>0.930</td>
<td>-1.380</td>
<td>0.976</td>
<td>38.60</td>
</tr>
<tr>
<td>9</td>
<td>0.938</td>
<td>-1.410</td>
<td>0.978</td>
<td>29.70</td>
</tr>
<tr>
<td>10</td>
<td>0.946</td>
<td>-1.410</td>
<td>0.983</td>
<td>23.40</td>
</tr>
<tr>
<td>11</td>
<td>0.956</td>
<td>-1.420</td>
<td>0.989</td>
<td>15.30</td>
</tr>
<tr>
<td>12</td>
<td>0.956</td>
<td>-1.430</td>
<td>0.986</td>
<td>10.60</td>
</tr>
<tr>
<td>13</td>
<td>0.956</td>
<td>-1.420</td>
<td>0.993</td>
<td>7.70</td>
</tr>
<tr>
<td>14</td>
<td>0.969</td>
<td>-1.420</td>
<td>0.997</td>
<td>5.77</td>
</tr>
<tr>
<td>15</td>
<td>0.964</td>
<td>-1.410</td>
<td>0.989</td>
<td>4.45</td>
</tr>
</tbody>
</table>

# Here we use the same power as was used for the human network to facilitate the comparison of #the two networks. Again, by playing around with different choices, one can easily verify that #our findings are highly robust with respect to the choice of the power.

correctChimp=correctHuman

## Calculation of the network (adjacency matrix) by raising the absolute value of the correlation #matrix to a power (soft-thresholding with the power adjacency function).

# Human network:

AdjMatHuman = abs(cor(datExpr[indexHuman,] ,use="p"))^correctHuman
diag(AdjMatHuman)=0

# Chimp network:

AdjMatChimp = abs(cor(datExpr[indexChimp,] ,use="p"))^correctChimp
diag(AdjMatChimp)=0

## Calculation of the whole network connectivity k:

ConnectivityHuman <- apply(AdjMatHuman,1,sum)  ConnectivityChimp <- apply(AdjMatChimp,1,sum)
## Depiction of scale-free topology:
# The black curve corresponds to scale-free topology and the red curve corresponds to truncated
#scale-free topology.

```r
par(mfrow=c(2,1))
ScaleFreePlot1(ConnectivityHuman, AF1=paste("Human ,","power=9"),truncated1=T)
ScaleFreePlot1(ConnectivityChimp, AF1=paste("Chimp ,","power=9"),truncated1=T)
```

## Scaling k to lie between 0 and 1:

```r
ConnectivityHuman=ConnectivityHuman/max(ConnectivityHuman)
ConnectivityChimp=ConnectivityChimp/max(ConnectivityChimp)
```

## Comparing gene expression human and chimp:

```r
ExpressionHuman=apply(datExpr[indexHuman,],2,mean)
ExpressionHuman=ExpressionHuman/max(ExpressionHuman)

ExpressionChimp=apply(datExpr[indexChimp,],2,mean)
ExpressionChimp=ExpressionChimp/max(ExpressionChimp)
```

```r
cor.test(ExpressionHuman,ExpressionChimp,method="s",use="p")
```
Output:

Spearman's rank correlation rho
data: ExpressionHuman and ExpressionChimp
S = 37835562, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.9644842

# Now we form a scatter plot between the chimp and human expression values:

par(mfrow=c(1,1))
plot(ExpressionHuman, ExpressionChimp, main="Cortex, caudate nucleus, cerebellum",
     xlab="Human gene expression", ylab="Chimp gene expression", sub="rho = 0.96",
     cex.main=2, cex.lab=2, cex.axis=1.5, cex.sub=1.25)
abline(lm(ExpressionChimp~ExpressionHuman), col="red", lwd=2)
## Comparing network connectivity between human and chimp:

cor.test(ConnectivityHuman, ConnectivityChimp, method="s", use="p")

Output
Spearman's rank correlation rho
data: ConnectivityHuman and ConnectivityChimp
S = 369361380, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
 rho
0.6537473

# Scatterplot between chimp and human connectivity:

par(mfrow=c(1,1))
plot(ConnectivityHuman, ConnectivityChimp, main="Cortex, caudate nucleus, cerebellum",
    xlab="Human network connectivity", ylab="Chimp network connectivity", sub="rho = 0.65",
cex.main=2, cex.lab=2, cex.axis=1.5, cex.sub=1.25)
abline(lm(ConnectivityChimp~ConnectivityHuman), col="red", lwd=2)
# As a pre-processing step towards module construction, we restrict the network to genes with reasonably high connectivity. This does not lead to a big loss of information since module genes tend to have high connectivity (7). Toward this end, consider the median connectivity in human and chimp:

\[
\text{median}(\text{ConnectivityHuman})
\]

0.09753368

# This motivates us to restrict the analysis to genes with \( k > 0.1 \) in either human or chimp:

\[
\text{minconnections} = .1
\]

rest1=ConnectivityChimp>minconnections | ConnectivityHuman>minconnections

\[
\text{table}(\text{rest1})
\]

AdjMatChimprest=AdjMatChimp[rest1,rest1]
AdjMatHumanrest=AdjMatHuman[rest1,rest1]
rm(AdjMatChimp);rm(AdjMatHuman); collect_garbage()

#Module Construction
# The topological overlap of two nodes reflects their similarity in terms of the commonality of the nodes they connect to, see (6, 10).

## Creating distance matrices based upon the topological overlap of 2241 genes for humans and chimpanzees:

\[
\text{distTOMChimp} \leftarrow \text{TOMdist1}(\text{AdjMatChimprest})
\]

\[
\text{distTOMHuman} \leftarrow \text{TOMdist1}(\text{AdjMatHumanrest})
\]

collect_garbage()

# To group genes with coherent expression profiles into modules, we use average linkage hierarchical clustering, which uses the topological overlap measure as dissimilarity.
## Performing average linkage hierarchical clustering using these distance matrices:

\[
\text{hierTOMHuman} \leftarrow \text{hclust}(\text{as.dist}(\text{distTOMHuman}), \text{method}="\text{average})"
\]

\[
\text{hierTOMChimp} \leftarrow \text{hclust}(\text{as.dist}(\text{distTOMChimp}), \text{method}="\text{average})"
\]
## Displaying dendrograms:

```r
par(mfrow = c(1, 2))
plot(hierTOMHuman, labels = F, main = "Human")
plot(hierTOMChimp, labels = F, main = "Chimp")
```

## Assign colors to modules based upon the height cutoff (h1) and minimum module size (minsize1):

# Once a dendrogram is obtained from a hierarchical clustering method, we need to choose a height cutoff h1 in order to arrive at a clustering. It is a judgement call where to cut the tree branches. The height cut-off can be found by inspection: a height cutoff value is chosen in the dendrogram such that some of the resulting branches correspond to the discrete diagonal blocks (modules) in the TOM plot.

```r
colorh1 = as.character(modulecolor2(hierTOMHuman, h1 = .95, minsize1 = 30))
table(colorh1)
```
# minsize specifies that a module should contain at least 30 genes.
par(mfrow=c(2,2),mar=c(2,2,2,2))
pplot(hierTOMHuman,main="Human brain",labels=F)
abline(h=.95,col="red")
pplot(hierTOMChimp,main="Chimpanzee brain",labels=F)
hclustplot1(hierTOMHuman,colorh1,title1="Human network, Human colors")
hclustplot1(hierTOMChimp,colorh1,title1="Chimp network, Human colors")

colorh=as.character(colorh1)
colorhALL=rep("grey", length(ConnectivityChimp))
colorhALL[rest1]=as.character(colorh)

## The topological overlap matrix (TOM) plot is a symmetric plot in which genes are ordered as
#they are depicted in the above dendrogram. The TOM plot depicts increasing topological
#overlap as increasing color intensity; therefore "squares" along the diagonal can be used to help
#define modules:
TOMplot1(distTOMHuman, hierTOMHuman , colorh)
## Creating a classical multi-dimensional scaling plots for humans and chimpanzees as another means of network representation:

```r
cmdChimp = cmdscale(as.dist(distTOMChimp), 3)
cmdHuman = cmdscale(as.dist(distTOMHuman), 3)
collect_garbage()
pairs(cmdHuman, col=as.character(colorh), main="Human network, Human colors", ylim=c(-.5,.4), xlim=c(-.55,.45))
```
pairs(cmdChimp, col=as.character(colorh), main="Chimp network, Human colors", ylim=c(-.5,.4), xlim=c(-.5,.45))

# Creating heatmaps for each module with samples ordered according to brain region:

byregionsdat1=c(20,26,32,2,8,14,21,27,33,3,9,15,22,28,34,4,10,16,23,29,35,5,11,17,24,30,36,6,12,18,25,31,37,7,13,19,26,32,38,8,14,20)
orderbyregion=c(1,2,3,4,5,6,1,2,3,4,5,6,1,2,3,4,5,6,1,2,3,4,5,6,1,2,3,4,5,6,1,2,3,4,5,6)
ordersamples2=order(orderbyregion)
datExprHumanrest=datExpr[indexHuman,rest1]
datExprChimprest=datExpr[indexChimp,rest1]

# Now we create a heatmap of the turquoise module genes (rows are genes, columns are microarray samples:
par(mar=c(2,3,4,3))
par(oma=c(0,0,2,0))
whichmodule="turquoise"
datcombined=data.frame(rbind(datExprHumanrest , datExprChimprest))[, colorh=whichmodule]
repvec=c(1,1,1,2,2,2)
labelHuman=as.character(rep(repvec,6))
par(mfrow=c(1,1))
plot.mat(t(scale(datcombined[ordersamples2,])), rlabels=dimnames(datcombined)[[2]], clabels=dimnames(dat1[byregionsdat1])[2], ccols=labelHuman, rcols=whichmodule)
par(mar=c(2,3,4,3))
par oma=c(0,0,2,0))
whichmodule="blue"
datcombined=data.frame(rbind(datExprHumanrest, datExprChimprest))[, colorh=whichmodule]
repvec=c(1,1,1,2,2,2)
labelHuman=as.character(rep(repvec,6))
par(mfrow=c(1,1))
plot.mat(t(scale(datcombined[ordersamples2,])) , rlabels= dimnames(datcombined)[2],
,clabels=dimnames(dat1[byregionsdat1][2]), ccols=labelHuman, rcols=whichmodule)


datcombined=data.frame(rbind(datExprHumanrest, datExprChimprest))[, colorh=whichmodule]
repvec=c(1,1,1,2,2,2)
labelHuman=as.character(rep(repvec,6))
par(mfrow=c(1,1))
plot.mat(t(scale(datcombined[ordersamples2,])), rlabels=dimnames(datcombined)[[2]], clabels=dimnames(dat1[byregionsdat1])[2], ccols=labelHuman, rcols=whichmodule)

par(mar=c(2,3,4,3))
par(oma=c(0,0,2,0))
whichmodule="yellow"
datcombined=data.frame(rbind(datExprHumanrest, datExprChimprest))[,, colorh==whichmodule]
repvec=c(1,1,1,2,2,2)
labelHuman=as.character(rep(repvec,6))
par(mfrow=c(1,1))
plot.mat(t(scale(datcombined[ordersamples2,])), rlabels=dimnames(datcombined)[[2]], clabels=dimnames(dat1[byregionsdat1])[2], ccols=labelHuman, rcols=whichmodule)

par(mar=c(2,3,4,3))
par(oma=c(0,0,2,0))
whichmodule="green"
datcombined=data.frame(rbind(datExprHumanrest, datExprChimprest))[,, colorh==whichmodule]
repvec=c(1,1,1,2,2,2)
labelHuman=as.character(rep(repvec,6))
par(mfrow=c(1,1))
plot.mat(t(scale(datcombined[ordersamples2,])), rlabels=dimnames(datcombined)[[2]], clabels=dimnames(dat1[byregionsdat1])[2], ccols=labelHuman, rcols=whichmodule)

par(mar=c(2,3,4,3))
par(oma=c(0,0,2,0))
whichmodule="red"
datcombined=data.frame(rbind(datExprHumanrest, datExprChimprest))[, colorh==whichmodule]
repvec=c(1,1,1,2,2,2)
labelHuman=as.character(rep(repvec,6))
par(mfrow=c(1,1))
plot.mat(t(scale(datcombined[ordersamples2,])), rlabels=dimnames(datcombined)[[2]], clabels=dimnames(dat1[byregionsdat1])[2], ccols=labelHuman, rcols=whichmodule)

par(mar=c(2,3,4,3))
par(oma=c(0,0,2,0))
whichmodule="black"
datcombined=data.frame(rbind(datExprHumanrest, datExprChimprest))[, colorh==whichmodule]
repvec=c(1,1,1,2,2,2)
labelHuman=as.character(rep(repvec,6))
par(mfrow=c(1,1))
# The module eigengenes are used to summarize the expression profiles in each module.
# To compute the module eigengene for each module, we use the first principal component
# estimated using the singular value decomposition:

PC1 = ModulePrinComps1(datExpr[1:36,rest1], colorh1)

# PC1 is a list with 2 components. The first component is a data frame whose columns
# correspond to the module eigengenes.

# Note that the column names list the modules (PC = first principal component = module
# eigengene)

names(PC1[[1]])
[1] "PCblack"  "PCblue"   "PCbrown"  "PCgreen"  "PCgrey"   "PCred"   
"PCturquoise" "PCyellow"

## Displaying the percent variance explained by the first five principal components for each
# module:

signif(PC1[[2]], 2)

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.680</td>
<td>0.470</td>
<td>0.530</td>
<td>0.450</td>
<td>0.270</td>
<td>0.620</td>
<td>0.560</td>
<td>0.630</td>
</tr>
<tr>
<td>2</td>
<td>0.100</td>
<td>0.120</td>
<td>0.094</td>
<td>0.140</td>
<td>0.150</td>
<td>0.093</td>
<td>0.093</td>
<td>0.081</td>
</tr>
<tr>
<td>3</td>
<td>0.039</td>
<td>0.063</td>
<td>0.084</td>
<td>0.076</td>
<td>0.120</td>
<td>0.049</td>
<td>0.042</td>
<td>0.056</td>
</tr>
<tr>
<td>4</td>
<td>0.032</td>
<td>0.051</td>
<td>0.049</td>
<td>0.045</td>
<td>0.094</td>
<td>0.033</td>
<td>0.037</td>
<td>0.038</td>
</tr>
<tr>
<td>5</td>
<td>0.029</td>
<td>0.044</td>
<td>0.032</td>
<td>0.033</td>
<td>0.056</td>
<td>0.026</td>
<td>0.030</td>
<td>0.031</td>
</tr>
</tbody>
</table>

# This table shows that the first principal component of the black module explains 68% of the
# variation. The second PC explains only 10% of the variation. For the improper grey module
# comprised of genes in no proper module, the first PC explains only 27% of the variation.
# Overall, this table justifies why the first PC can be used to sum up the behavior of the proper modules.

## Displaying and exporting the values of the first module eigengenes for all samples and all modules (used for barplots in Figure 4):

```R
PC1[[1]]
write.table(PC1[[1]], file="First PC.csv", sep="", row.names=dimnames(datExpr)[[1]][1:36])
dimnames(datExpr)[[1]]
```

## To characterize modules by assigning an indicator variable based upon the first module eigengene. This indicator variable is 1 for cerebellum samples and 0 otherwise:

### Turquoise:
```R
cereb=rep(c(0,0,0,0,0,1),6)
PC1turquoise=PC1[[1]]$PCturquoise
```

### Now we study whether the module eigengene is differentially expressed between cerebellar and non-cerebellar samples
```R
kruskal.test(PC1turquoise, cereb)
```

Output
```
Kruskal-Wallis rank sum test
data:  PC1turquoise and cereb
Kruskal-Wallis chi-squared = 14.5946, df = 1, p-value = 0.0001333
```

# To visualize this result, one can also use the following boxplot:
```R
par(mfrow=c(1,1))
boxplot(split(PC1turquoise, cereb), notch=T, varwidth=T, ylab="Turquoise Module Eigengene Expression", xlab="cerebellum(=1) versus other regions")
```

# Message: the expression values of the turquoise module genes are highly determined by whether or not the microarray sample is the cerebellum or not.

```r
# Blue
cortical = rep(c(1,1,1,1,0,0),6)
PC1blue = PC1[[1]]$PCblue
kruskal.test(PC1blue, cortical)

# Brown
PC1brown = PC1[[1]]$PCbrown
kruskal.test(PC1brown, cortical)

# Yellow
caudate = rep(c(0,0,0,0,1,0),6)
PC1yellow = PC1[[1]]$PCyellow
kruskal.test(PC1yellow, caudate)

# Green
cerebcort = rep(c(1,1,1,1,0,1),6)
PC1green = PC1[[1]]$PCgreen
kruskal.test(PC1green, cerebcort)

# Red
caudacc = rep(c(0,1,0,0,1,0),6)
PC1red = PC1[[1]]$PCred
kruskal.test(PC1red, caudacc)
```

```r
datExpr2 = data.frame(t(dat1[Brain_variant_H>0,1:39]))
datExpr3 = t(datExpr2)
dimnames(datExpr3)[[2]]
```

```r
# To calculate within module connectivity ($k_{in}$) for all genes in each module for human and chimp:

## Black
AdjMatChimprestblack = AdjMatChimprest[colorh1 == "black", colorh1 == "black"]
ConnectivityChimpblack = apply(AdjMatChimprestblack, 1, sum)
AdjMatHumanrestblack = AdjMatHumanrest[colorh1 == "black", colorh1 == "black"]
ConnectivityHumanblack = apply(AdjMatHumanrestblack, 1, sum)
ConnectivityChimpblack = ConnectivityChimpblack / max(ConnectivityChimpblack)
ConnectivityHumanblack = ConnectivityHumanblack / max(ConnectivityHumanblack)

## Blue
AdjMatChimprestblue = AdjMatChimprest[colorh1 == "blue", colorh1 == "blue"]
ConnectivityChimpblue = apply(AdjMatChimprestblue, 1, sum)
AdjMatHumanrestblue = AdjMatHumanrest[colorh1 == "blue", colorh1 == "blue"]
ConnectivityHumanblue = apply(AdjMatHumanrestblue, 1, sum)
ConnectivityChimpblue = ConnectivityChimpblue / max(ConnectivityChimpblue)
ConnectivityHumanblue = ConnectivityHumanblue / max(ConnectivityHumanblue)

## Brown
AdjMatChimprestbrown = AdjMatChimprest[colorh1 == "brown", colorh1 == "brown"]
ConnectivityChimbpbrown = apply(AdjMatChimprestbrown, 1, sum)
AdjMatHumanrestbrown = AdjMatHumanrest[colorh1 == "brown", colorh1 == "brown"]
```
ConnectivityHumanbrown=apply(AdjMatHumanrestbrown,1,sum)

ConnectivityChimpbrown=ConnectivityChimpbrown/max(ConnectivityChimpbrown)
ConnectivityHumanbrown=ConnectivityHumanbrown/max(ConnectivityHumanbrown)

#Green

AdjMatChimprestgreen=AdjMatChimprest[colorh1=="green",colorh1=="green"]
ConnectivityChimpgreen=apply(AdjMatChimprestgreen,1,sum)
AdjMatHumanrestgreen=AdjMatHumanrest[colorh1=="green",colorh1=="green"]
ConnectivityHumangreen=apply(AdjMatHumanrestgreen,1,sum)

ConnectivityChimpgreen=ConnectivityChimpgreen/max(ConnectivityChimpgreen)
ConnectivityHumangreen=ConnectivityHumangreen/max(ConnectivityHumangreen)

#Red

AdjMatChimprestred=AdjMatChimprest[colorh1=="red",colorh1=="red"]
ConnectivityChimpred=apply(AdjMatChimprestred,1,sum)
AdjMatHumanrestred=AdjMatHumanrest[colorh1=="red",colorh1=="red"]
ConnectivityHumanred=apply(AdjMatHumanrestred,1,sum)

ConnectivityChimpred=ConnectivityChimpred/max(ConnectivityChimpred)
ConnectivityHumanred=ConnectivityHumanred/max(ConnectivityHumanred)

#Turquoise

AdjMatChimprestturquoise=AdjMatChimprest[colorh1=="turquoise",colorh1=="turquoise"]
ConnectivityChimpturquoise=apply(AdjMatChimprestturquoise,1,sum)
AdjMatHumanrestturquoise=AdjMatHumanrest[colorh1=="turquoise",colorh1=="turquoise"]
ConnectivityHumanturquoise=apply(AdjMatHumanrestturquoise,1,sum)

ConnectivityChimpturquoise=ConnectivityChimpturquoise/max(ConnectivityChimpturquoise)
ConnectivityHumanturquoise=ConnectivityHumanturquoise/max(ConnectivityHumanturquoise)

#Yellow

AdjMatChimprestyellow=AdjMatChimprest[colorh1=="yellow",colorh1=="yellow"]
ConnectivityChimpyellow=apply(AdjMatChimprestyellow,1,sum)
AdjMatHumanrestryellow=AdjMatHumanrest[colorh1=="yellow",colorh1=="yellow"]
ConnectivityHumanyellow=apply(AdjMatHumanrestryellow,1,sum)

ConnectivityChimpyellow=ConnectivityChimpyellow/max(ConnectivityChimpyellow)
ConnectivityHumanyellow=ConnectivityHumanyellow/max(ConnectivityHumanyellow)

## To compare the correlation in $k_{in}$ between humans and chimpanzees by module:

```
cor.test(ConnectivityHumanblack, ConnectivityChimpblack, method="s")
cor.test(ConnectivityHumanblue, ConnectivityChimpblue, method="s")
cor.test(ConnectivityHumanbrown, ConnectivityChimpbrown, method="s")
cor.test(ConnectivityHumangreen, ConnectivityChimpgreen, method="s")
cor.test(ConnectivityHumanred, ConnectivityChimpred, method="s")
cor.test(ConnectivityHumanturquoise, ConnectivityChimpturquoise, method="s")
cor.test(ConnectivityHumanyellow, ConnectivityChimpyellow, method="s")
```
## To calculate the topological overlap for all genes within each module for human and chimp:

### Black

```r
distTOMHumanblack <- TOMdist1(AdjMatHumanrestblack)
simTOMHumanblack = 1-distTOMHumanblack
diag(simTOMHumanblack)=0

distTOMChimpblack <- TOMdist1(AdjMatChimprestblack)
simTOMChimpblack = 1-distTOMChimpblack
diag(simTOMChimpblack)=0
```

### Blue

```r
distTOMHumanblue <- TOMdist1(AdjMatHumanrestblue)
simTOMHumanblue = 1-distTOMHumanblue
diag(simTOMHumanblue)=0

distTOMChimpblue <- TOMdist1(AdjMatChimprestblue)
simTOMChimpblue = 1-distTOMChimpblue
diag(simTOMChimpblue)=0
```

### Brown

```r
distTOMHumanbrown <- TOMdist1(AdjMatHumanrestbrown)
simTOMHumanbrown = 1-distTOMHumanbrown
diag(simTOMHumanbrown)=0

distTOMChimpbrown <- TOMdist1(AdjMatChimprestbrown)
simTOMChimpbrown = 1-distTOMChimpbrown
diag(simTOMChimpbrown)=0
```

### Green

```r
distTOMHumangreen <- TOMdist1(AdjMatHumanrestgreen)
simTOMHumangreen = 1-distTOMHumangreen
diag(simTOMHumangreen)=0

distTOMChimpgreen <- TOMdist1(AdjMatChimprestgreen)
simTOMChimpgreen = 1-distTOMChimpgreen
diag(simTOMChimpgreen)=0
```

### Red

```r
distTOMHumanred <- TOMdist1(AdjMatHumanrestred)
simTOMHumanred = 1-distTOMHumanred
diag(simTOMHumanred)=0

distTOMChimpred <- TOMdist1(AdjMatChimprestred)
simTOMChimpred = 1-distTOMChimpred
diag(simTOMChimpred)=0
```

### Turquoise

```r
distTOMHumanturquoise <- TOMdist1(AdjMatHumanrestturquoise)
simTOMHumanturquoise = 1-distTOMHumanturquoise
diag(simTOMHumanturquoise)=0

distTOMChimpturquoise <- TOMdist1(AdjMatChimprestturquoise)
simTOMChimpturquoise = 1-distTOMChimpturquoise
diag(simTOMChimpturquoise)=0
```
# Yellow

distTOMHumanyellow <- TOMdist1(AdjMatHumanrestyellow)
simTOMHumanyellow = 1-distTOMHumanyellow
diag(simTOMHumanyellow)=0

distTOMChimpyellow <- TOMdist1(AdjMatChimprestyellow)
simTOMChimpyellow = 1-distTOMChimpyellow
diag(simTOMChimpyellow)=0

## Creating a matrix for each module to describe the relative topological overlap for all
# connections between human and chimp:

# Black

simRatioblack=simTOMHumanblack/mean(simTOMHumanblack)/(simTOMHumanblack/mean(simTOMHum
anblack)+ simTOMChimpblack/mean(simTOMChimpblack)+0.00001)

# Comment: we add the negligible constant of 0.00001 to the denominator as a pre-caution to
# ensure that the denominator is non-zero.

# Blue

simRatioblue=simTOMHumanblue/mean(simTOMHumanblue)/(simTOMHumanblue/mean(simTOMHumanbl
ue)+ simTOMChimpblue/mean(simTOMChimpblue)+0.00001)

# Brown

simRatiobrown=simTOMHumanbrown/mean(simTOMHumanbrown)/(simTOMHumanbrown/mean(simTOMHum
anbrown)+ simTOMChimpbrown/mean(simTOMChimpbrown)+0.00001)

# Green

simRatiogreen=simTOMHumangreen/mean(simTOMHumangreen)/(simTOMHumangreen/mean(simTOMHum
angreen)+ simTOMChimpgreen/mean(simTOMChimpgreen)+0.00001)

# Red

simRatiored=simTOMHumanred/mean(simTOMHumanred)/(simTOMHumanred/mean(simTOMHumanred)+
 simTOMChimpred/mean(simTOMChimpred)+0.00001)

# Turquoise

simRatioturquoise=simTOMHumanturquoise/mean(simTOMHumanturquoise)/(simTOMHumanturquois
e/mean(simTOMHumanturquoise) + simTOMChimpturquoise/mean(simTOMChimpturquoise)+0.00001)

# Yellow

simRatiroyellow=simTOMHumanyellow/mean(simTOMHumanyellow)/(simTOMHumanyellow/mean(simTO
MHumanyellow) + simTOMChimpyellow/mean(simTOMChimpyellow)+0.00001)

## Comparing gene expression and k between human and chimp for networks comprised of
# different brain regions (Datasets 2-4).

## Without caudate nucleus:
dat2=read.csv("Dataset 2 (no CN).csv",header=T)
attach(dat2)
dim(dat2)
datExpr2=data.frame(t(dat2[Brain_variant_H>0,2:32]))
dim(datExpr2)
dimnames(datExpr2)[[1]]
indexHuman=16:30
indexChimp=1:15
powerHuman=9
powerChimp=9
AdjMatHuman2 = abs(cor(datExpr2[indexHuman,],use="p"))^{powerHuman}
AdjMatChimp2 = abs(cor(datExpr2[indexChimp,],use="p"))^{powerChimp}
diag(AdjMatHuman2)=0
diag(AdjMatChimp2)=0
ConnectivityHuman2 <- apply(AdjMatHuman2,1,sum)
ConnectivityChimp2 <- apply(AdjMatChimp2,1,sum)
ConnectivityHuman2=ConnectivityHuman2/max(ConnectivityHuman2)
ConnectivityChimp2=ConnectivityChimp2/max(ConnectivityChimp2)
ExpressionHuman2=apply(datExpr2[indexHuman,],2,mean)
ExpressionHuman2=ExpressionHuman2/max(ExpressionHuman2)
ExpressionChimp2=apply(datExpr2[indexChimp,],2,mean)
ExpressionChimp2=ExpressionChimp2/max(ExpressionChimp2)
cor.test(ExpressionHuman2,ExpressionChimp2,method="s",use="p")

par(mar=c(5.1,5.1,4.1,2.1))
plot(ExpressionHuman2,ExpressionChimp2,main="Cortex, cerebellum", xlab="Human gene expression", ylab="Chimp gene expression", sub="rho = 0.96", cex.main=2, cex.lab=2, cex.axis=1.5, cex.sub=1.25)
abline(lm(ExpressionChimp2~ExpressionHuman2),col="red",lwd=2)

cor.test(ConnectivityHuman2,ConnectivityChimp2,method="s",use="p")
par(mar=c(5.1,5.1,4.1,2.1))
plot(ConnectivityHuman2,ConnectivityChimp2,main="Cortex, cerebellum", xlab="Human network connectivity", ylab="Chimp network connectivity", sub="rho = 0.65", cex.main=2, cex.lab=2, cex.axis=1.5, cex.sub=1.25)
abline(lm(ConnectivityChimp2~ConnectivityHuman2),col="red",lwd=2)

###Without cerebellum:

dat3=read.csv("Dataset 3 (no CB).csv",header=T)
attach(dat3)
dim(dat3)
datExpr3=data.frame(t(dat3[Brain_variant_H>0,2:32]))
dim(datExpr3)
dimnames(datExpr3)[[1]]
indexHuman=c(16:30)
indexChimp=c(1:15)
powerHuman=9
powerChimp=9
AdjMatHuman3 = abs(cor(datExpr3[indexHuman,] ,use="p"))^{powerHuman}
AdjMatChimp3 = abs(cor(datExpr3[indexChimp,] ,use="p"))^{powerChimp}
diag(AdjMatHuman3)=0
diag(AdjMatChimp3)=0
ConnectivityHuman3 <- apply(AdjMatHuman3,1,sum)
ConnectivityChimp3 <- apply(AdjMatChimp3,1,sum)
ConnectivityHuman3=ConnectivityHuman3/max(ConnectivityHuman3)
ConnectivityChimp3=ConnectivityChimp3/max(ConnectivityChimp3)
ExpressionHuman3=apply(datExpr3[indexHuman,],2,mean)
ExpressionChimp3=apply(datExpr3[indexChimp,],2,mean)
ExpressionChimp3=ExpressionChimp3/max(ExpressionChimp3)
cor.test(ExpressionHuman3,ExpressionChimp3,method="s",use="p")

par(mar=c(5.1,5.1,4.1,2.1))
plot(ExpressionHuman3,ExpressionChimp3,main="Cortex, caudate nucleus", xlab="Human gene expression", ylab="Chimp gene expression", sub="rho = 0.96", cex.main=2, cex.lab=2, cex.axis=1.5, cex.sub=1.25)
abline(lm(ExpressionChimp3~ExpressionHuman3),col="red",lwd=2)
cor.test(ConnectivityHuman3,ConnectivityChimp3,method="s",use="p")

par(mar=c(5.1,5.1,4.1,2.1))
plot(ConnectivityHuman3,ConnectivityChimp3,main="Cortex, caudate nucleus", xlab="Human network connectivity", ylab="Chimp network connectivity", sub="rho = 0.51", cex.main=2, cex.lab=2, cex.axis=1.5, cex.sub=1.25)
abline(lm(ConnectivityChimp3~ConnectivityHuman3),col="red",lwd=2)

##Without caudate nucleus and cerebellum:

dat4=read.csv("Dataset 4 (no CN or CB).csv",header=T)
attach(dat4)
dim(dat4)
datExpr4=data.frame(t(dat4[Brain_variant_H>0,2:26]))
dim(datExpr4)
dimnames(datExpr4)[[1]]
indexHuman=c(13:24)
indexChimp=c(1:12)
powerHuman=9
\[\text{powerChimp} = 9\]

\[\text{AdjMatHuman4} = \text{abs}(\text{cor(datExpr4[indexHuman,], use="p")))}^{\text{powerHuman}}\]

\[\text{AdjMatChimp4} = \text{abs}(\text{cor(datExpr4[indexChimp,], use="p")))}^{\text{powerChimp}}\]

diag(AdjMatHuman4)=0

diag(AdjMatChimp4)=0

ConnectivityHuman4 <- apply(AdjMatHuman4,1,sum)

ConnectivityChimp4 <- apply(AdjMatChimp4,1,sum)

ConnectivityHuman4=ConnectivityHuman4/max(ConnectivityHuman4)

ConnectivityChimp4=ConnectivityChimp4/max(ConnectivityChimp4)

ExpressionHuman4=apply(datExpr4[indexHuman,],2,mean)

ExpressionChimp4=apply(datExpr4[indexChimp,],2,mean)

ExpressionHuman4=ExpressionHuman4/max(ConnectivityHuman4)

ExpressionChimp4=ExpressionChimp4/max(ConnectivityChimp4)

cor.test(ExpressionHuman4,ExpressionChimp4,method="s",use="p")

par(mar=c(5.1,5.1,4.1,2.1))
plot(ExpressionHuman4,ExpressionChimp4,main="Cortex", xlab="Human gene expression", ylab="Chimp gene expression", sub="rho = 0.96", cex.main=2, cex.lab=2, cex.axis=1.5, cex.sub=1.25)
abline(lm(ExpressionChimp4~ExpressionHuman4),col="red",lwd=2)

cor.test(ConnectivityHuman4,ConnectivityChimp4,method="s",use="p")

par(mar=c(5.1,5.1,4.1,2.1))
plot(ConnectivityHuman4,ConnectivityChimp4,main="Cortex", xlab="Human network connectivity", ylab="Chimp network connectivity", sub="rho = 0.22", cex.main=2, cex.lab=2, cex.axis=1.5, cex.sub=1.25)
abline(lm(ConnectivityChimp4~ConnectivityHuman4),col="red",lwd=2)
Identification of a module corresponding to primary visual cortex in a dataset lacking cerebellar samples (Dataset 3):

dat1=read.csv("Dataset 3 (no CB).csv",header=T)
attach(dat1)
dim(dat1)
datExpr=data.frame(t(dat1[Brain_variant_H>0,2:32]))
dim(datExpr)
dimnames(datExpr)[[1]]
indexChimp=c(1:15)
indexHuman=c(16:30)
powerHuman=9
powerChimp=9
collect_garbage()
AdjMatHuman = abs(cor(datExpr[indexHuman,],use="p"))^{powerHuman}
AdjMatChimp = abs(cor(datExpr[indexChimp,],use="p"))^{powerChimp}
diag(AdjMatHuman)=0
diag(AdjMatChimp)=0
ConnectivityHuman <- apply(AdjMatHuman,1,sum)
ConnectivityChimp <- apply(AdjMatChimp,1,sum)
ConnectivityHuman=ConnectivityHuman/max(ConnectivityHuman)
ConnectivityChimp=ConnectivityChimp/max(ConnectivityChimp)
minconnections=.1
rest1=ConnectivityChimp>minconnections | ConnectivityHuman>minconnections
table(rest1)

AdjMatChimprest=AdjMatChimp[rest1,rest1]
AdjMatHumanrest=AdjMatHuman[rest1,rest1]
rm(AdjMatChimp);rm(AdjMatHuman); collect_garbage()
distTOMChimp <- TOMdist1(AdjMatChimprest )
distTOMHuman <- TOMdist1(AdjMatHumanrest )
collect_garbage()

hierTOMHuman <- hclust(as.dist(distTOMHuman),method="average")
hierTOMChimp <- hclust(as.dist(distTOMChimp),method="average")

colorh1=as.character(modulocolor2(hierTOMHuman,h1=.98,minsize1=30))
table(colorh1)
par(mfrow=c(2,2),mar=c(2,2,2,2))
plot(hierTOMHuman, main="Human brain", labels=F)
abline(h=.98, col="red")
plot(hierTOMChimp, main="Chimpanzee brain", labels=F)
hclustplot1(hierTOMHuman, colorh1, title1="Human network, Human colors")
hclustplot1(hierTOMChimp, colorh1, title1="Chimp network, Human colors")

colorh=as.character(colorh1)

# Heat map for primary visual cortex module:

par(mar=c(2,3,4,3))
par(oma=c(0,0,2,0))
byregionsdat1=c(17,22,27,2,7,12,18,23,28,3,8,13,19,24,29,4,9,14,20,25,30,5,10,15,21,26,31,6,11,16)
orderbyregion=c(1,2,3,4,5,1,2,3,4,5,1,2,3,4,5,1,2,3,4,5,1,2,3,4,5)
ordersamples2=order(orderbyregion)
datExprHumanrest=datExpr[indexHuman,rest1]
datExprChimprest=datExpr[indexChimp,rest1]

whichmodule="brown"
datcombined=data.frame(rbind(datExprHumanrest , datExprChimprest))[, colorh==whichmodule]
repvec=c(1,1,1,2,2,2)
labelHuman=as.character(rep(repvec,5))
par(mfrow=c(1,1))
plot.mat(t(scale(datcombined[ordersamples2,])) , rlabels= dimnames(datcombined)[[2]], clabels= dimnames(dat1[byregionsdat1])[2]], ccols=labelHuman, rcols=whichmodule)

## To obtain the principal components of each module by singular value decomposition:

PC1=ModulePrinComps1(datExpr[1:30,rest1],colorh1)

## Displaying the percent variance explained by the first five principal components for each module:

PC1[[2]]
names(PC1[[1]])

## Note: X2 = brown module = primary visual cortex module.
## Displaying and exporting the values of the first module eigengenes for all samples in the
#primary visual cortex module (used for barplot in Figure 4):

PC1[[1]]

write.table(PC1[[1]], file="First PC.csv", sep="",
row.names=dimnames(datExpr)[[1]][1:30])

## To characterize the primary visual cortex module by assigning an indicator variable based
#upon the first module eigengene:

# Brown
vision=rep(c(0,0,1,0,0),6)
PC1brown=PC1[[1]]$PCbrown
kruskal.test(PC1brown,vision)

## To calculate within module connectivity ($k_{in}$) for all genes visual module:

datExpr2=data.frame(t(dat1[Brain_variant_H>0,1:32]))
datExpr3=t(datExpr2)
dimnames(datExpr3)[[2]]

colorh=as.character(colorh1)
colorhALL=rep("grey", length(ConnectivityChimp))
colorhALL[rest1]=as.character(colorh)

# Visual

AdjMatChimprestvisual=AdjMatChimprest[colorh1=="brown",colorh1=="brown"]
ConnectivityChimpvisual=apply(AdjMatChimprestvisual,1,sum)
AdjMatHumanrestvisual=AdjMatHumanrest[colorh1=="brown",colorh1=="brown"]
ConnectivityHumanvisual=apply(AdjMatHumanrestvisual,1,sum)

ConnectivityChimpvisual=ConnectivityChimpvisual/max(ConnectivityChimpvisual)
ConnectivityHumanvisual=ConnectivityHumanvisual/max(ConnectivityHumanvisual)

## To compare the correlation in kin between human and chimpanzee for visual module:

cor.test(ConnectivityHumanvisual, ConnectivityChimpvisual, method="s")

## To calculate the topological overlap for all genes within the visual module for human and
chimp:

# Visual

distTOMHumanvisual <- TOMdist1(AdjMatHumanrestvisual)
simTOMHumanvisual = 1-distTOMHumanvisual
diag(simTOMHumanvisual)=0
distTOMChimpvisual <- TOMdist1(AdjMatChimprestvisual)
simTOMChimpvisual = 1-distTOMChimpvisual
diag(simTOMChimpvisual)=0

## Creating a matrix to describe the relative topological overlap for all connections between
#human and chimp:

# Visual
simRatiovisual = simTOMHumanvisual/mean(simTOMHumanvisual)/(simTOMHumanvisual/mean(simTOMHumanvisual)+ simTOMChimpvisual/mean(simTOMChimpvisual)+0.00001)

THE END