Simulation Studies and R code for studying the relationship between heterogeneity and a soft threshold used for a weighted network construction

Steve Horvath, Jun Dong
Correspondence: shorvath@mednet.ucla.edu

In Horvath and Dong (2008), we mention that the heterogeneity of most weighted gene co-expression networks increases with the soft threshold beta. Recall that a weighted co-expression network is defined by raising a co-expression similarity (e.g. the absolute correlation) to a power beta:

\[ a_{ij} = s_{ij}^{\text{beta}}. \]

Recall that we define the network heterogeneity as coefficient of variation of the connectivity (degree) vector \( k \), i.e. we define

\[ \text{Heterogeneity} = \frac{\sqrt{\text{variance}(k)}}{\text{mean}(k)} = \frac{nS_2(k)}{\sqrt{S_1(k)^2}} - 1. \]

The node connectivity is given by

\[ \text{Connectivity}_i = k_i = \sum_{j \neq i} a_{ij}. \]

In the following, we describe a simulation study that can be used to argue that for the vast majority of networks, the heterogeneity increases with the soft threshold beta. Thus, for most co-expression networks, increasing beta makes it easier to discern highly connected genes from less connected genes. However, we will also show that one can construct networks for which increasing beta leads to a lower heterogeneity.

Reference


The following R software code allows one to show that the heterogeneity of a network with beta=1 tends to be lower than that of a network with beta>1.

```r
# The number of genes in the network is specified by the following
no.genes=100
# the number of random networks is given in the following
no.replicates=20000
# This vector will contain the heterogeneity for the network with beta=1
heterogeneity1=rep(NA, no.replicates)
heterogeneitybeta=rep(NA, no.replicates)

# this vector specifies random values of the soft threshold beta larger than 1
betavector=sample(2:20, no.replicates, replace=T)
```
for (i in 1:no.replicates) {
    # here we set a random seed
    set.seed(i)
    # next we define a similarity measure with entries in [0,1]
    SIMILARITY=matrix(runif(no.genes*no.genes,min=0, max=1), ncol=no.genes, nrow=no.genes)
    # The following ensures that the similarity is symmetric.
    SIMILARITY=.5*SIMILARITY+.5*t(SIMILARITY)
    # Since the connectivity is defined without regard to the diagonal element, we
    # find it convenient to set the diagonal to 0.
    diag(SIMILARITY)=0
    # The following is the connectivity vector for a network with beta=1
    k1=as.numeric(apply(SIMILARITY^1,2,sum))
    vark1=var(k1)
    meank1=mean(k1)
    # let’s avoid dividing by zero
    if (meank1==0 ) heterogeneity1[i]=0
    if (meank1>0 ) heterogeneity1[i]=sqrt(vark1)/mean(k1)
    # this is the connectivity vector for a network with beta>1
    kbeta= as.numeric(apply(SIMILARITY^betavector[i],2,sum))
    varkbeta=var(kbeta)
    meankbeta=mean(kbeta)
    if (meankbeta==0 ) heterogeneitybeta[i]=0
    if (meankbeta>0 ) heterogeneitybeta[i]=sqrt(varkbeta)/mean(kbeta)
}

# Now we summarize our findings

table(heterogeneity1<=heterogeneitybeta)
mean(heterogeneity1<=heterogeneitybeta)

Output
> table(heterogeneity1<=heterogeneitybeta)
  TRUE
20000
> mean(heterogeneity1<=heterogeneitybeta)
[1] 1

Message: In all of the 20000 comparison, we find that increasing beta leads to a higher heterogeneity. One needs to look really hard to find a network where increasing beta leads to a lower heterogeneity. Toward this end, we find it useful to construct a network with very few genes as is done in the following code.
# Looking for similarity measure for which increasing beta does not lead to higher heterogeneity.

# Here we define a tiny network comprised of few genes
no.genes=5
# the number of random networks is given in the following
no.replicates=20000
# This vector will contain the heterogeneity for the network with beta=1
heterogeneity1=rep(NA, no.replicates)
heterogeneitybeta=rep(NA, no.replicates)
#this vector specifies the soft thresholds beta equal to 2
betavector=rep(2, no.replicates)
for (i in 1:no.replicates) {
    # here we set a random seed
    set.seed(i)
    # next we define a similarity measure with entries in [0,1]
    SIMILARITY=matrix(runif(no.genes*no.genes,min=0, max=1), ncol=no.genes, nrow=no.genes)
    # The following ensures that the similarity is symmetric.
    SIMILARITY=.5*SIMILARITY+.5*t(SIMILARITY)
    # Since the connectivity is defined without regard to the diagonal element, we
    # find it convenient to set the diagonal to 0.
    diag(SIMILARITY)=0
    # The following is the connectivity vector for a network with beta=1
    k1=as.numeric(apply(SIMILARITY^1,2,sum))
    vark1=var(k1)
    meank1=mean(k1)
    # let’s avoid dividing by zero
    if (meank1==0 ) heterogeneity1[i]=0
    if (meank1>0 ) heterogeneity1[i]=sqrt(vark1)/mean(k1)
    # this is the connectivity vector for a network with beta>1
    kbeta= as.numeric(apply(SIMILARITY^betavector[i],2,sum))
    varkbeta=var(kbeta)
    meankbeta=mean(kbeta)
    if (meankbeta==0 ) heterogeneitybeta[i]=0
    if (meankbeta>0 ) heterogeneitybeta[i]=sqrt(varkbeta)/mean(kbeta)
}

table(heterogeneity1<=heterogeneitybeta)
mean(heterogeneity1<=heterogeneitybeta)
Output
> table(heterogeneity1<=heterogeneitybeta)
FALSE TRUE
 41  19959
> mean(heterogeneity1<=heterogeneitybeta)
[1] 0.99795

Message: Only in 41 out of the 20000 tiny networks does increasing beta lead to a lower heterogeneity.

Let’s find one of these networks.

which(heterogeneity1>heterogeneitybeta)
> which(heterogeneity1>heterogeneitybeta)
[1]  1378  1630  1859  2311  3494  3952  3989  4174  4609  5416  6062
 6281
[13]  6309  7061  7112  7728  8389  9056  9532 10322 10440 10747
[25] 11264 11800 11958 12635 13547 13594 14048 14052 15414 15692 16229
 16387
[37] 16800 17512 17666 18491 19373

index.exception=1378

Note that
data.frame(heterogeneity1,heterogeneitybeta)[ index.exception,]

                heterogeneity1 heterogeneitybeta
index.exception 0.05711143        0.04534454

Here is the corresponding similarity matrix:

set.seed(index.exception)
SIMILARITY=matrix(runif(no.genes*no.genes,min=0, max=1), ncol=no.genes, nrow=no.genes)
SIMILARITY=.5*SIMILARITY+.5*t(SIMILARITY)
SIMILARITY
> SIMILARITY
[1,] 0.43963480 0.06389987 0.5941163 0.7843168 0.7536465
[2,] 0.06389987 0.48470342 0.9023037 0.4898024 0.5679353
[3,] 0.59411626 0.90230365 0.1119732 0.4898024 0.6865423
[4,] 0.78431680 0.48980241 0.3276598 0.1244695 0.3473306
[5,] 0.75364650 0.56793535 0.3473306 0.6865423 0.9914276

Discussion
Unlike the heterogeneity, the eigengene-based heterogeneity always increases with beta. For a proof see the methods section in Horvath and Dong (2008).