```r
RNAseq = RNAseq[apply(RNAseq,1,function(x) sum(x==0))<ncol(RNAseq)*0.8,]
library(limma)
RNAseq_voom = voom(RNAseq)$E

#transpose matrix to correlate genes in the following
WGCNA_matrix = t(RNAseq_voom[order(apply(RNAseq_voom,1,mad), decreasing = T)[1:5000],])

#similarity measure between gene profiles: biweight midcorrelation
library(WGCNA)
s = abs(bicor(WGCNA_matrix))
powers = c(c(1:10), seq(from = 12, to=20, by=2))
sft = pickSoftThreshold(WGCNA_matrix, powerVector = powers, verbose = 5)
plot(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
     xlab='Soft Threshold (power)',ylab='Scale Free Topology Model Fit,signed R^2',
     type='n', main = paste('Scale independence'));
text(sft$fitIndices[,1], -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
     labels=powers,cex=1,col='red'); abline(h=0.90,col='red')

#calculation of adjacency matrix
beta = 3
a = s^beta

#dissimilarity measure
w = 1-a

#create gene tree by average linkage hierarchical clustering
geneTree = hclust(as.dist(w), method = 'average')

#module identification using dynamic tree cut algorithm
modules = cutreeDynamic(dendro = geneTree, distM = w, deepSplit = 4, pamRespectsDendro = FALSE,
                        minClusterSize = 30)
#assign module colours
module.colours = labels2colors(modules)

#plot the dendrogram and corresponding colour bars underneath
plotDendroAndColors(geneTree, module.colours, 'Module colours', dendroLabels = FALSE, hang =
0.03,
                    addGuide = TRUE, guideHang = 0.05, main='')
library(ape)

#calculate eigengenes
MEs = moduleEigengenes(WGCNA_matrix, colors = module.colours, excludeGrey = FALSE)$eigengenes

#calculate dissimilarity of module eigengenes
MEDiss = 1-cor(MEs);

#cluster module eigengenes
METree = hclust(as.dist(MEDiss), method = 'average');

#plot the result with phytools package
par(mar=c(2,2,2,2))
plot.phylo(as.phylo(METree),type = 'fan',show.tip.label = FALSE, main='')
tiplabels(frame = 'circle',col='black', text=rep('',length(unique(modules))), bg =
levels(as.factor(module.colours)))

#load clinical metadata. Make sure that patient barcodes are in the same format
#create second expression matrix for which the detailed clinical data is available
WGCNA_matrix2 = WGCNA_matrix[match(clinical$Name, rownames(WGCNA_matrix)),]

#CAVE: 1 sample of detailed clinical metadata is not in downloaded data (TCGA-GN-A269-01')
not.available = which(is.na(rownames(WGCNA_matrix2))==TRUE)
WGCNA_matrix2 = WGCNA_matrix2[-not.available,]
str(WGCNA_matrix2)

#hence it needs to be removed from clinical table for further analysis
clinical = clinical[-not.available,]
```

```
#grouping in high and low lymphocyte score (lscore)
lscore = as.numeric(clinical$LYMPHOCYTE.SCORE)
lscore[lscore<3] = 0
lscore[lscore>0] = 1

#calculate gene significance measure for lymphocyte score (lscore) - Welch's t-Test
GS_lscore =
t(sapply(1:ncol(WGCNA_matrix2),function(x)c(t.test(WGCNA_matrix2[,x]~lscore,var.equal=F)$p.value,


t.test(WGCNA_matrix2[,x]~lscore,var.equal=F)$estimate[1],

t.test(WGCNA_matrix2[,x]~lscore,var.equal=F)$estimate[2])))
GS_lscore = cbind(GS.lscore, abs(GS_lscore[,2] - GS_lscore[,3]))
colnames(GS_lscore) = c('p_value','mean_high_lscore','mean_low_lscore',
                        'effect_size(high-low score)'); rownames(GS_lscore) =
colnames(WGCNA_matrix2)

#reference genes = all 5000 top mad genes
ref_genes = colnames(WGCNA_matrix2)

#create data frame for GO analysis
library(org.Hs.eg.db)
GO = toTable(org.Hs.egGO); SYMBOL = toTable(org.Hs.egSYMBOL)
GO_data_frame = data.frame(GO$go_id,
GO$Evidence,SYMBOL$symbol[match(GO$gene_id,SYMBOL$gene_id)])

#create GOAllFrame object
library(AnnotationDbi)
GO_ALLFrame = GOAllFrame(GOFrame(GO_data_frame, organism = 'Homo sapiens'))

#create gene set
library(GSEABase)
gsc <- GeneSetCollection(GO_ALLFrame, setType = GOCollection())

#perform GO enrichment analysis and save results to list - this make take several minutes
library(GEOstats)
GSEAGO = vector('list',length(unique(modules)))
for(i in 0:(length(unique(modules))-1)){
  GSEAGO[[i+1]] = summary(hyperGTest(GSEAGOHyperGParams(name = 'Homo sapiens GO',
                                                        geneSetCollection = gsc, geneIds =
colnames(RNAseq)[modules==i],
                                                        universeGeneIds = ref.genes, ontology =
'BP', pvalueCutoff = 0.05,
                                                        conditional = FALSE, testDirection =
'over')))
  print(i)
}

cutoff_size = 100

GO_module_name = rep(NA,length(unique(modules)))
for (i in 1:length(unique(modules))){
  GO.module.name[i] =
    GSEAGO[[i]][GSEAGO[[i]]$Size<cutoff_size,
                ][which(GSEAGO[[i]][GSEAGO[[i]]$Size<cutoff_size,]$Count==max(GSEAGO[[i]]
[GSEAGO[[i]]$

Size<cutoff.size,]$Count)),7]
}

GO.module.name[1] = 'module 0'

#calculate module significance
MS.lscore = as.data.frame(cbind(GS.lscore,modules))
MS.lscore$log_p_value = -log10(as.numeric(MS.lscore$p_value))
MS.lscore = ddply(MS.lscore, .(modules), summarize, mean(log_p_value), sd(log_p_value))
```

```
colnames(MS.lscore) = c('modules','pval','sd')
MS.lscore.bar = as.numeric(MS.lscore[,2])
MS.lscore.bar[MS.lscore.bar<(-log10(0.05))] = 0
names(MS.lscore.bar) = GO.module.name

METree.GO = METree
label.order = match(METree$labels,paste0('ME',labels2colors(0:(length(unique(modules))-1))))
METree.GO$labels = GO.module.name[label.order]
plotTree.wBars(as.phylo(METree.GO), MS.lscore.bar, tip.labels = TRUE, scale = 0.2)

#Calculate module membership
MM = abs(bicor(RNAseq, MEs))

#plot individual module of interest (MOI)
MOI = 3 #T cell differentiation co-expression module
plot(-log10(GS.lscore[modules==MOI,1]), MM[modules==MOI,MOI], pch=20,
     cex=(GS.lscore[modules==MOI,4]/max(GS.lscore[,4],na.rm=TRUE))*4,
     xlab='p-value (-log10) lymphocyte score', ylab='membership to module 3')
abline(v=-log10(0.05), lty=2, lwd=2)
sessionInfo()

## R version 3.4.0 (2017-04-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.2 LTS
##
## Matrix products: default
## BLAS: /home/biocbuild/bbs-3.6-bioc/R/lib/libRblas.so
## LAPACK: /home/biocbuild/bbs-3.6-bioc/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] RTCGAToolbox_2.7.0 BiocStyle_2.5.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.10      knitr_1.15.1      magrittr_1.5
##  [4] splines_3.4.0     lattice_0.20-35   stringr_1.2.0
##  [7] tools_3.4.0       grid_3.4.0        data.table_1.10.4
## [10] htmltools_0.3.5   yaml_2.1.14       survival_2.41-3
## [13] rprojroot_1.2     digest_0.6.12     RJSONIO_1.3-0
## [16] Matrix_1.2-9      bitops_1.0-6      RCurl_1.95-4.8
## [19] evaluate_0.10     rmarkdown_1.4     limma_3.33.0
## [22] stringi_1.1.5     compiler_3.4.0    RCircos_1.2.0
## [25] backports_1.0.5   XML_3.98-1.6
```