main

July 12, 2021

1 WGCNA

- @author = 'Apua Paquola'
- Edits by K.J. Benjamin
- Edits2 by Arthur S. Feltrin
  - New scale-free plots, export data to create network on cytoscape/igraph and format for jupyter notebook (05/2019)
  - Conversion from Rscript to jupyter notebook

Final edits by K.J. Benjamin for publication

[1]:
```
PARAM_NETWORK_TYPE = 'signed'
```

1.1 Prepare Data and Traits Table

[2]:
```
filter_outliers = function(expression, z_threshold = 2.5)
{
    # Input: an expression matrix
    # Output: an expression matrix with outliers removed
    # Remove samples with z normalized total distance from other samples >␣
 ↪z_threshold

    sample_distance = dist(expression)
    dist_z = scale(colSums(as.matrix(sample_distance)))
    stopifnot(all(rownames(dist_z) == rownames(expression)))

    keepSamples = dist_z < z_threshold

    new_expression = expression[keepSamples,]
    new_expression
}
```

[3]:
```
prepare_data=function()
{
  suppressMessages(library(dplyr))
  # Load sample data
  load("../../../differential_analysis/dentateGyrus/_m/genes/voomSVA.RData")
  sample_table = v$design %>% as.data.frame %>% select(-Intercept) %>%
```

```
    rename("Ancestry"="EA", "Sex"="Male")

  # Load residualized expression
  vsd <- data.table::fread(paste0("../../../differential_analysis/dentateGyrus/
↪",
                                  "_m/genes/residualized_expression.tsv")) %>%
    replace(is.na(.), "") %>% tibble::column_to_rownames("V1")
  print(dim(vsd))

  # Keep only the columns and rows that are present in
  # both the sample table and vsd file
  samples = intersect(colnames(vsd), rownames(sample_table))
  vsd = vsd[,samples]
  sample_table = sample_table[samples,]

  # WGCNA data import
  suppressMessages(library(WGCNA))
  options(stringsAsFactors = FALSE)
  datExpr0 = t(vsd)

  # Remove offending genes and samples from the data
  gsg = goodSamplesGenes(datExpr0, verbose = 3);
  if (!gsg$allOK)
  {
    datExpr0 = datExpr0[gsg$goodSamples, gsg$goodGenes]
  }
  datExpr=datExpr0
  # Remove outliers
  datExpr = filter_outliers(datExpr0, z_threshold = 2.5)
  rm(datExpr0)
  # Clean data
  samples = intersect(rownames(datExpr), rownames(sample_table))
  sample_table = sample_table[samples,]
  datExpr = datExpr[samples,]
  print(dim(datExpr))
  save(datExpr, sample_table, file = '00.RData')
}
```

1.2 Create Sample Dendogram Based on Distance (h)

[4]:
```
prepare_traits = function()
{
    lnames = load('00.RData')
    # Associate traits with samples
    traitRows = match(rownames(datExpr), rownames(sample_table))
    datTraits = sample_table[traitRows,]
    # Diagnostic plot: Sample dendrogram and trait heatmap
```

```
    pdf(file='sample_dendrogram_and_trait_heatmap.pdf',height=16,width = 22)
    sampleTree2 = hclust(dist(datExpr), method = "average")
    # Convert traits to a color representation: white means
    # low, red means high, grey means missing entry
    traitColors = numbers2colors(traitRows, signed=FALSE);
    # Plot the sample dendrogram and the colors underneath.
    plotDendroAndColors(sampleTree2, traitColors, groupLabels="Avg. Counts",
                        main = "Sample dendrogram and trait heatmap",
                        cex.dendroLabels=0.7)
    dev.off()
    # Print output
    plotDendroAndColors(sampleTree2, traitColors, groupLabels="Avg. Counts",
                        main = "Sample dendrogram and trait heatmap",
                        cex.dendroLabels=0.75)
    save(datExpr, sample_table, datTraits, file = "01.RData")
}
```

1.3 Calculate Scale-Free Topology

[5]:
```
plot_power_parameter=function(datExpr, plot_filename)
{
    # Choose a set of soft-thresholding powers
    powers = seq(from = 1, to=30, by=1)
    # Call the network topology analysis function
    sft = pickSoftThreshold(datExpr, networkType = PARAM_NETWORK_TYPE,
                            powerVector = powers, verbose = 5)
    # Plot the results:
    pdf(file=plot_filename)
    par(mfcol = c(2,2));
    par(mar = c(4.2, 4.5 , 2.2, 0.5),oma=c(0,0,2,0))
    cex1 = 0.7;
    # Scale-free topology fit index as a function of the
    # soft-thresholding power
    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=cex1,col="blue");
    # this line corresponds to using an R^2 cut-off of h
    abline(h=0.80,col="red")
    # Mean connectivity as a function of the soft-thresholding power
    plot(sft$fitIndices[,1], sft$fitIndices[,5],
         xlab="Soft Threshold (power)", ylab="Mean Connectivity",
         type="n", main = paste("Mean connectivity"))
    text(sft$fitIndices[,1], sft$fitIndices[,5], labels=powers,␣
↪cex=cex1,col="blue")
```

```
    #####
    plot(sft$fitIndices[,1], sft$fitIndices[,6],
         xlab="Soft Threshold (power)", ylab="Median Connectivity",
         type="n", main = paste("Median connectivity"))
    text(sft$fitIndices[,1], sft$fitIndices[,6], labels=powers,␣
↪cex=cex1,col="blue")
    #####
    plot(sft$fitIndices[,1], sft$fitIndices[,7],
         xlab="Soft Threshold (power)", ylab="Max Connectivity",
         type="n", main = paste("Max connectivity"))
    text(sft$fitIndices[,1], sft$fitIndices[,7], labels=powers,␣
↪cex=cex1,col="blue")
    dev.off()
    ####plot on jupyter
    par(mfcol = c(2,2));
    par(mar = c(4.2, 4.5 , 2.2, 0.5),oma=c(0,0,2,0))
    cex1 = 0.7;
    # Scale-free topology fit index as a function of the
    # soft-thresholding power
    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=cex1,col="blue");
    # this line corresponds to using an R^2 cut-off of h
    abline(h=0.80,col="red")
    # Mean connectivity as a function of the soft-thresholding power
    plot(sft$fitIndices[,1], sft$fitIndices[,5],
         xlab="Soft Threshold (power)", ylab="Mean Connectivity",
         type="n", main = paste("Mean connectivity"))
    text(sft$fitIndices[,1], sft$fitIndices[,5], labels=powers,␣
↪cex=cex1,col="blue")
    #####
    plot(sft$fitIndices[,1], sft$fitIndices[,6],
         xlab="Soft Threshold (power)", ylab="Median Connectivity",
         type="n", main = paste("Median connectivity"))
    text(sft$fitIndices[,1], sft$fitIndices[,6], labels=powers,␣
↪cex=cex1,col="blue")
    #####
    plot(sft$fitIndices[,1], sft$fitIndices[,7],
         xlab="Soft Threshold (power)",ylab="Max Connectivity", type="n",
         main = paste("Max connectivity"))
    text(sft$fitIndices[,1], sft$fitIndices[,7], labels=powers,␣
↪cex=cex1,col="blue")
}
```

[6]:
```
figure_out_power_parameter=function()
{
    library(WGCNA)
    options(stringsAsFactors = FALSE);
    #enableWGCNAThreads(nThreads=16)
    lnames = load(file = '01.RData')
    plot_power_parameter(datExpr, 'power_parameter_selection.pdf')
}
```

1.4 Build the Network

[7]:
```
construct_network=function()
{
  library(WGCNA)
  options(stringsAsFactors = FALSE);
  enableWGCNAThreads(nThreads=16)
  lnames = load(file = "01.RData")

  # softPower value from previous plot power_parameter_selection.pdf
  softPower = 9; # Based on the Hippocampus
  # ALWAYS choose a value equal or above (better) 0.8
  cor <- WGCNA::cor
  net = blockwiseModules(datExpr, power = softPower,
                         networkType = PARAM_NETWORK_TYPE,
                         TOMType = PARAM_NETWORK_TYPE,
                         numericLabels = TRUE,
                         corType = "bicor", mergeCutHeight = 0.2,
                         saveTOMs = TRUE, saveTOMFileBase = "TOM",
                         verbose = 3, maxBlockSize=30000)

  moduleLabels = net$colors
  moduleColors = labels2colors(net$colors)
  MEs = net$MEs;
  geneTree = net$dendrograms[[1]];
  save(net, MEs, moduleLabels, moduleColors, geneTree, softPower, file = "02.
 ↪RData")
}
#cyt = exportNetworkToCytoscape(modTOM,
```

1.5 Use Topology Overlap Matrix (TOM) to cluster the genes on the networks into different modules

[8]:
```
plot_cluster_dendrogram=function()
{
    library(WGCNA)
    options(stringsAsFactors = FALSE);
    enableWGCNAThreads(nThreads=16)
```

```
    load(file = "02.RData")
    pdf(file="cluster_dendrogram.pdf",height=16,width = 22)
    mergedColors = labels2colors(net$colors)
    plotDendroAndColors(net$dendrograms[[1]], mergedColors[net$blockGenes[[1]]],
                        "Module Colors", dendroLabels = FALSE, hang = 0.03,
                        addGuide = TRUE, guideHang = 0.05, cex.dendroLabels=0.3)
    dev.off()
    # Print output
    plotDendroAndColors(net$dendrograms[[1]], mergedColors[net$blockGenes[[1]]],
                      "Module Colors", dendroLabels = FALSE, hang = 0.03,
                      addGuide = TRUE, guideHang = 0.05, cex.dendroLabels=0.3)
}
```

1.6 Use Pearson Correlation to measure the correlation between each module eigenvalue (kME) and the various sample traits

[9]:
```
correlate_with_traits=function()
{
    library(WGCNA)
    options(stringsAsFactors = FALSE)
    enableWGCNAThreads(nThreads=16)
    lnames = load(file = "01.RData")
    lnames = load(file = "02.RData")
    # Define numbers of genes and samples
    nGenes = ncol(datExpr);
    nSamples = nrow(datExpr);
    # Recalculate MEs with color labels
    MEs0 = moduleEigengenes(datExpr, moduleColors)$eigengenes
    MEs = orderMEs(MEs0)
    moduleTraitCor = cor(MEs, datTraits, use = "p");
    moduleTraitPvalue = corPvalueStudent(moduleTraitCor, nSamples);
    # Plot
    pdf(file="module_trait_relationships.pdf", height=22,width = 26)
    # Will display correlations and their p-values
    textMatrix = paste(signif(moduleTraitCor, 2), "\n(",
                       signif(moduleTraitPvalue, 1), ")", sep = "");
    dim(textMatrix) = dim(moduleTraitCor)
    par(mar = c(6, 8.5, 3, 3));
    # Display the correlation values within a heatmap plot
    labeledHeatmap(Matrix = moduleTraitCor,
                   xLabels = names(datTraits),
                   yLabels = names(MEs),
                   ySymbols = names(MEs),
                   colorLabels = FALSE,
                   naColor = "grey",
                   colors = blueWhiteRed(50),
                   textMatrix = textMatrix,
```

```
                    setStdMargins = FALSE,
                    cex.text = 0.9,
                    zlim = c(-1,1),
                    main = paste("Module kME-Trait Correlation"))
    dev.off()
    # Print output
    textMatrix = paste(signif(moduleTraitCor, 2), "\n(",
                       signif(moduleTraitPvalue, 1), ")", sep = "");
    dim(textMatrix) = dim(moduleTraitCor)
    par(mar = c(12, 6.5, 3, 0.5));
    # Display the correlation values within a heatmap plot
    labeledHeatmap(Matrix = moduleTraitCor, xLabels = names(datTraits),
                    yLabels = names(MEs), ySymbols = names(MEs),
                    colorLabels = FALSE, naColor = "grey",
                    colors = blueWhiteRed(50), textMatrix = textMatrix,
                    setStdMargins = FALSE, cex.text = 0.55, zlim = c(-1,1),
                    main = paste("Module kME-Trait Correlation"))
}
```

1.7 Export the main results

[10]:
```
export_eigengene_tables = function()
{
    library(WGCNA)
    options(stringsAsFactors = FALSE)
    lnames = load(file = "01.RData")
    lnames = load(file = "02.RData")
    # Define numbers of genes and samples
    nGenes = ncol(datExpr)
    nSamples = nrow(datExpr)
    # Recalculate MEs with color labels
    MEs0 = moduleEigengenes(datExpr, moduleColors)$eigengenes
    rownames(MEs0) = rownames(datExpr)
    write.csv(MEs0, 'eigengenes.csv')
    # Write modules
    modules = data.frame(row.names=colnames(datExpr), module=moduleColors)
    write.csv(modules, 'modules.csv')
    save(datExpr,softPower,moduleColors, file = "cytoscapenetwork.Rdata")
}
```

1.8 Run the functions and plot the results

[11]:
```
prepare_data()
```

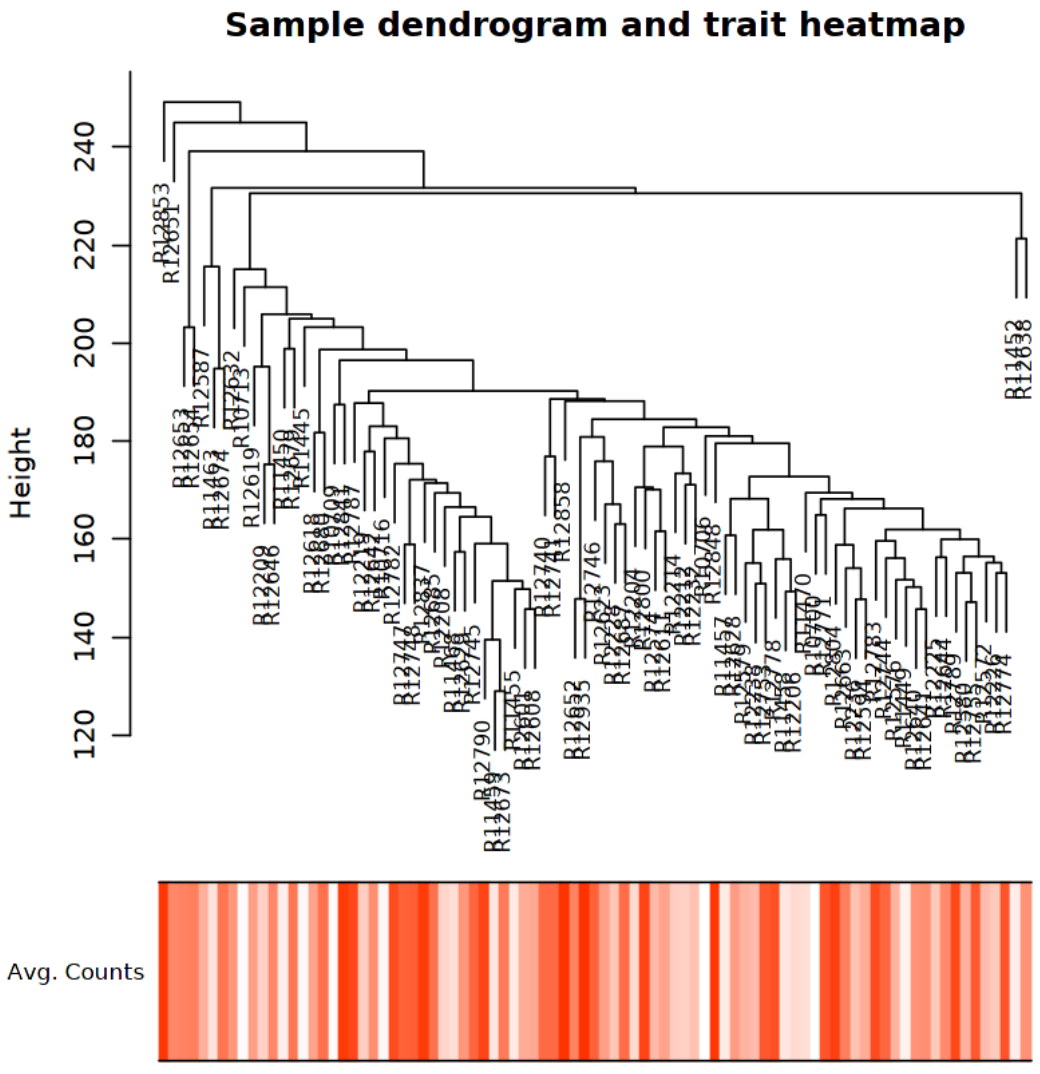
```
Loading required package: limma

Warning message in
```

```
data.table::fread(paste0("../../../differential_analysis/dentateGyrus/", :
"Detected 90 column names but the data has 91 columns (i.e. invalid file). Added
1 extra default column name for the first column which is guessed to be row
names or an index. Use setnames() afterwards if this guess is not correct, or
fix the file write command that created the file to create a valid file."

[1] 21140    90
 Flagging genes and samples with too many missing values…
  ..step 1
[1]    87 21140
```

[12]:
```
# 1 - Sample dendrogram and trait heatmap
prepare_traits()
```



[13]:
```
# 2 - Scale Free Topology Model Fit
figure_out_power_parameter()
construct_network()
```

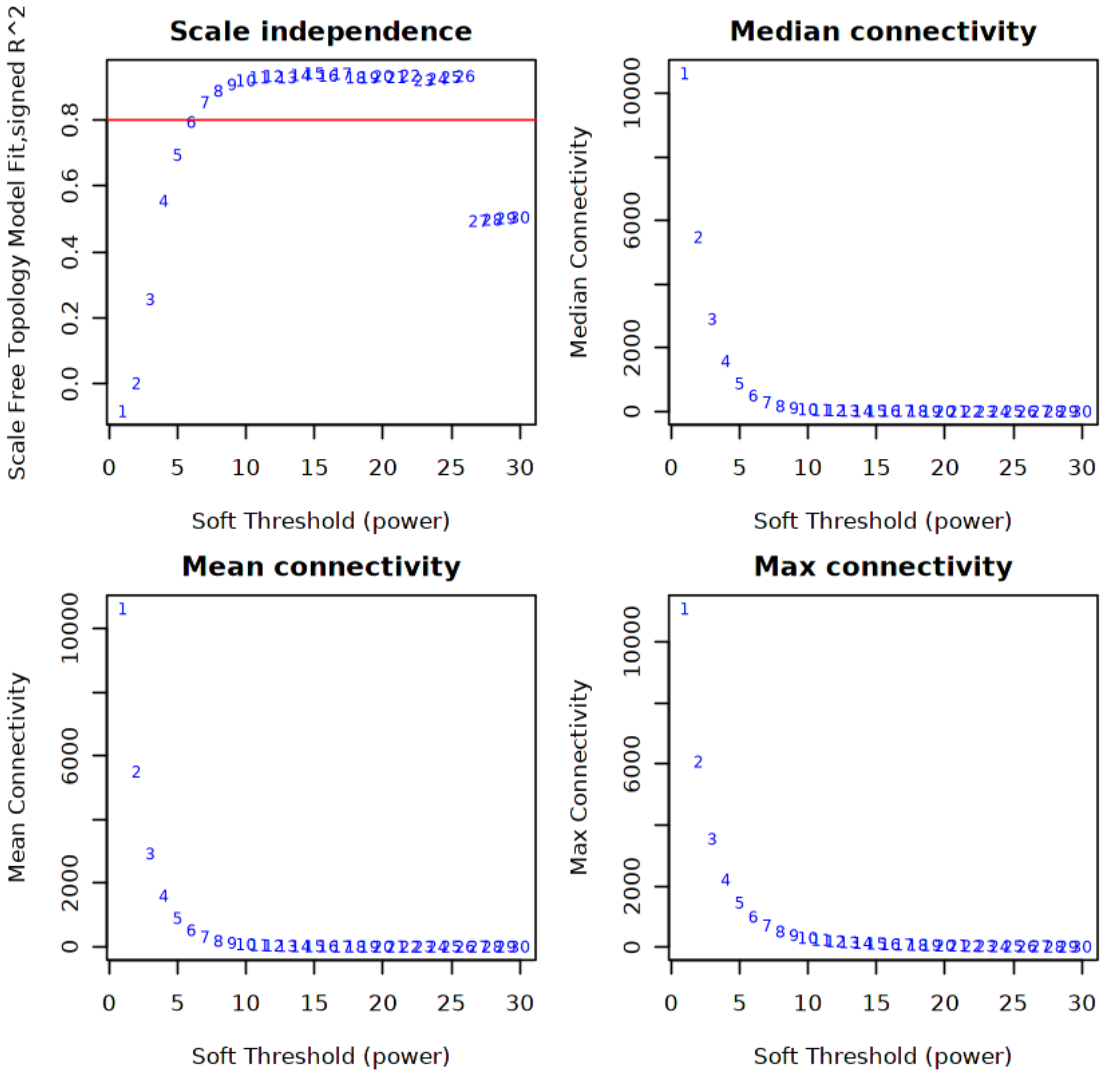
```
pickSoftThreshold: will use block size 2116.
 pickSoftThreshold: calculating connectivity for given powers…
   ..working on genes 1 through 2116 of 21140
```

```
Warning message:
"executing %dopar% sequentially: no parallel backend registered"
```

```
   ..working on genes 2117 through 4232 of 21140
   ..working on genes 4233 through 6348 of 21140
   ..working on genes 6349 through 8464 of 21140
   ..working on genes 8465 through 10580 of 21140
   ..working on genes 10581 through 12696 of 21140
   ..working on genes 12697 through 14812 of 21140
   ..working on genes 14813 through 16928 of 21140
   ..working on genes 16929 through 19044 of 21140
   ..working on genes 19045 through 21140 of 21140
   Power SFT.R.sq slope truncated.R.sq  mean.k. median.k.  max.k.
1      1  0.08230 24.90          0.996 1.06e+04  1.06e+04 11100.0
2      2  0.00201 -1.50          0.954 5.50e+03  5.48e+03  6080.0
3      3  0.25700 -8.21          0.901 2.93e+03  2.90e+03  3540.0
4      4  0.55400 -8.17          0.932 1.60e+03  1.57e+03  2210.0
5      5  0.69300 -6.69          0.955 8.94e+02  8.71e+02  1450.0
6      6  0.79300 -5.69          0.974 5.13e+02  4.92e+02   992.0
7      7  0.85300 -4.94          0.986 3.01e+02  2.83e+02   704.0
8      8  0.88700 -4.40          0.992 1.81e+02  1.66e+02   516.0
9      9  0.90700 -4.02          0.994 1.11e+02  9.90e+01   390.0
10    10  0.91900 -3.73          0.996 6.96e+01  6.00e+01   304.0
11    11  0.92800 -3.48          0.996 4.46e+01  3.68e+01   241.0
12    12  0.93200 -3.29          0.996 2.91e+01  2.29e+01   195.0
13    13  0.92900 -3.16          0.994 1.94e+01  1.45e+01   160.0
14    14  0.93600 -2.99          0.994 1.32e+01  9.25e+00   133.0
15    15  0.94100 -2.85          0.996 9.13e+00  5.97e+00   112.0
16    16  0.93400 -2.78          0.993 6.43e+00  3.89e+00    95.2
17    17  0.94000 -2.67          0.997 4.60e+00  2.57e+00    81.6
18    18  0.92800 -2.62          0.993 3.35e+00  1.72e+00    70.5
19    19  0.92800 -2.55          0.993 2.47e+00  1.16e+00    61.3
20    20  0.93300 -2.46          0.994 1.85e+00  7.87e-01    53.6
21    21  0.93000 -2.41          0.993 1.40e+00  5.39e-01    47.1
22    22  0.93400 -2.34          0.994 1.08e+00  3.73e-01    41.6
23    23  0.92000 -2.31          0.981 8.36e-01  2.60e-01    36.9
24    24  0.92500 -2.25          0.983 6.56e-01  1.83e-01    32.9
25    25  0.92900 -2.20          0.985 5.20e-01  1.29e-01    29.4
26    26  0.93200 -2.14          0.985 4.16e-01  9.21e-02    26.4
27    27  0.49400 -2.70          0.500 3.35e-01  6.58e-02    23.8
28    28  0.49700 -2.65          0.502 2.73e-01  4.73e-02    21.5
```

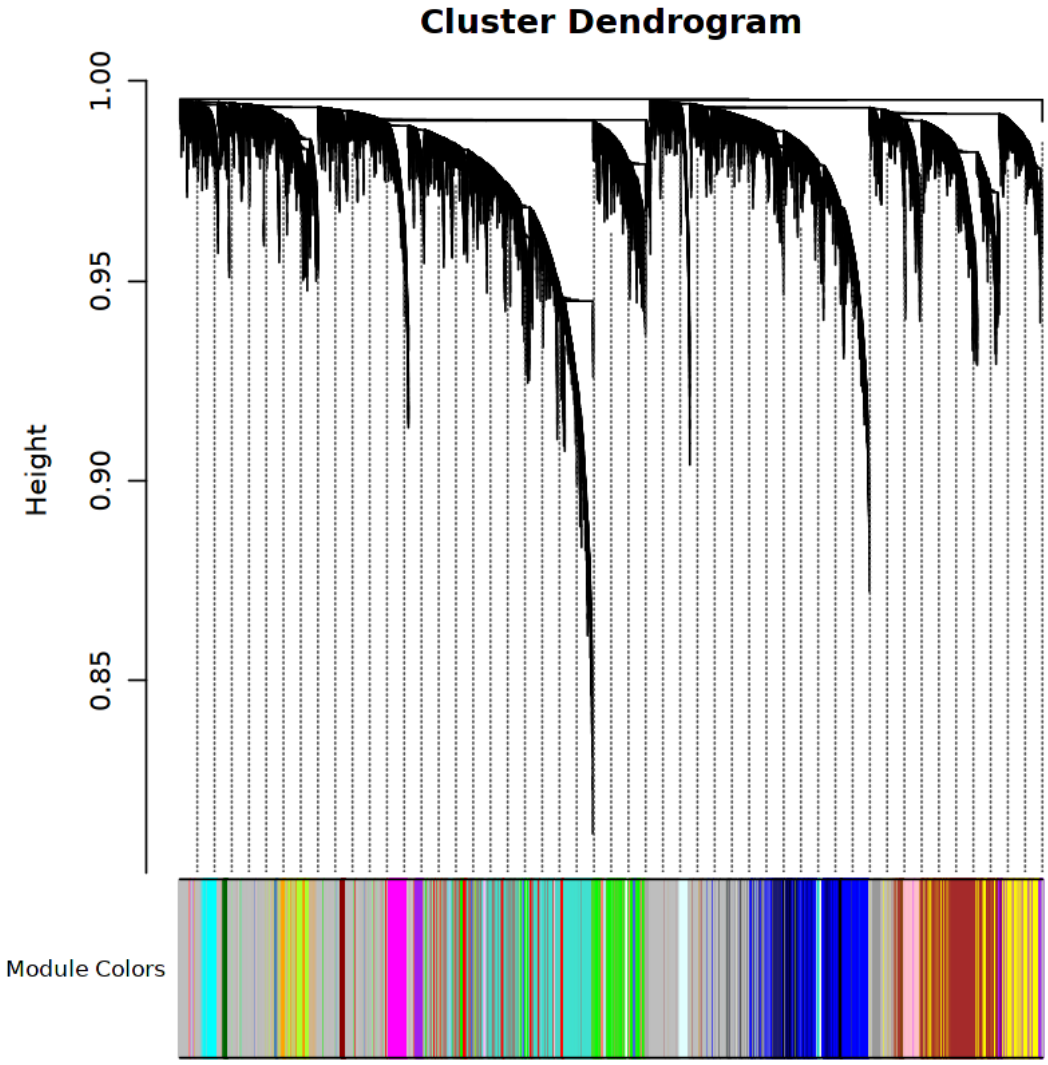
```
29    29  0.50200 -2.60          0.508 2.23e-01  3.41e-02    19.4
30    30  0.50500 -2.55          0.510 1.84e-01  2.47e-02    17.6
Allowing parallel execution with up to 16 working processes.
 Calculating module eigengenes block-wise from all genes
   Flagging genes and samples with too many missing values…
    ..step 1
 ..Working on block 1 .
    TOM calculation: adjacency..
    ..will use 16 parallel threads.
     Fraction of slow calculations: 0.000000
    ..connectivity..
    ..matrix multiplication (system BLAS)..
    ..normalization..
    ..done.
   ..saving TOM for block 1 into file TOM-block.1.RData
 …clustering..
 …detecting modules..
 …calculating module eigengenes..
 …checking kME in modules..
     ..removing 1496 genes from module 1 because their KME is too low.
     ..removing 1005 genes from module 2 because their KME is too low.
     ..removing 791 genes from module 3 because their KME is too low.
     ..removing 347 genes from module 4 because their KME is too low.
     ..removing 440 genes from module 5 because their KME is too low.
     ..removing 179 genes from module 6 because their KME is too low.
     ..removing 298 genes from module 7 because their KME is too low.
     ..removing 8 genes from module 8 because their KME is too low.
     ..removing 412 genes from module 9 because their KME is too low.
     ..removing 33 genes from module 10 because their KME is too low.
     ..removing 2 genes from module 11 because their KME is too low.
     ..removing 124 genes from module 12 because their KME is too low.
     ..removing 82 genes from module 13 because their KME is too low.
     ..removing 37 genes from module 15 because their KME is too low.
     ..removing 153 genes from module 16 because their KME is too low.
     ..removing 1 genes from module 17 because their KME is too low.
     ..removing 5 genes from module 18 because their KME is too low.
     ..removing 64 genes from module 19 because their KME is too low.
     ..removing 46 genes from module 20 because their KME is too low.
     ..removing 25 genes from module 21 because their KME is too low.
     ..removing 66 genes from module 22 because their KME is too low.
     ..removing 17 genes from module 23 because their KME is too low.
     ..removing 47 genes from module 24 because their KME is too low.
     ..removing 40 genes from module 25 because their KME is too low.
     ..removing 1 genes from module 26 because their KME is too low.
     ..removing 1 genes from module 27 because their KME is too low.
     ..removing 10 genes from module 28 because their KME is too low.
     ..removing 12 genes from module 29 because their KME is too low.
     ..removing 1 genes from module 30 because their KME is too low.
```

```
    ..removing 13 genes from module 31 because their KME is too low.
    ..removing 15 genes from module 32 because their KME is too low.
    ..removing 1 genes from module 35 because their KME is too low.
    ..removing 1 genes from module 36 because their KME is too low.
    ..removing 1 genes from module 37 because their KME is too low.
    ..removing 1 genes from module 41 because their KME is too low.
    ..removing 13 genes from module 43 because their KME is too low.
    ..removing 1 genes from module 45 because their KME is too low.
    ..removing 2 genes from module 46 because their KME is too low.
    ..removing 2 genes from module 48 because their KME is too low.
    ..removing 1 genes from module 53 because their KME is too low.
    ..removing 2 genes from module 57 because their KME is too low.
 ..reassigning 80 genes from module 1 to modules with higher KME.
 ..reassigning 43 genes from module 2 to modules with higher KME.
 ..reassigning 20 genes from module 3 to modules with higher KME.
 ..reassigning 1 genes from module 4 to modules with higher KME.
 ..reassigning 11 genes from module 6 to modules with higher KME.
 ..reassigning 4 genes from module 7 to modules with higher KME.
 ..reassigning 1 genes from module 8 to modules with higher KME.
 ..reassigning 4 genes from module 9 to modules with higher KME.
 ..reassigning 7 genes from module 10 to modules with higher KME.
 ..reassigning 1 genes from module 11 to modules with higher KME.
 ..reassigning 5 genes from module 12 to modules with higher KME.
 ..reassigning 3 genes from module 13 to modules with higher KME.
 ..reassigning 6 genes from module 14 to modules with higher KME.
 ..reassigning 2 genes from module 15 to modules with higher KME.
 ..reassigning 1 genes from module 16 to modules with higher KME.
 ..reassigning 6 genes from module 18 to modules with higher KME.
 ..reassigning 1 genes from module 20 to modules with higher KME.
 ..reassigning 2 genes from module 22 to modules with higher KME.
 ..reassigning 5 genes from module 23 to modules with higher KME.
 ..reassigning 1 genes from module 24 to modules with higher KME.
 ..reassigning 1 genes from module 28 to modules with higher KME.
 ..reassigning 1 genes from module 32 to modules with higher KME.
 ..reassigning 1 genes from module 33 to modules with higher KME.
 ..reassigning 2 genes from module 37 to modules with higher KME.
 ..reassigning 1 genes from module 41 to modules with higher KME.
 ..reassigning 1 genes from module 55 to modules with higher KME.
..merging modules that are too close..
    mergeCloseModules: Merging modules whose distance is less than 0.2
      Calculating new MEs…
```



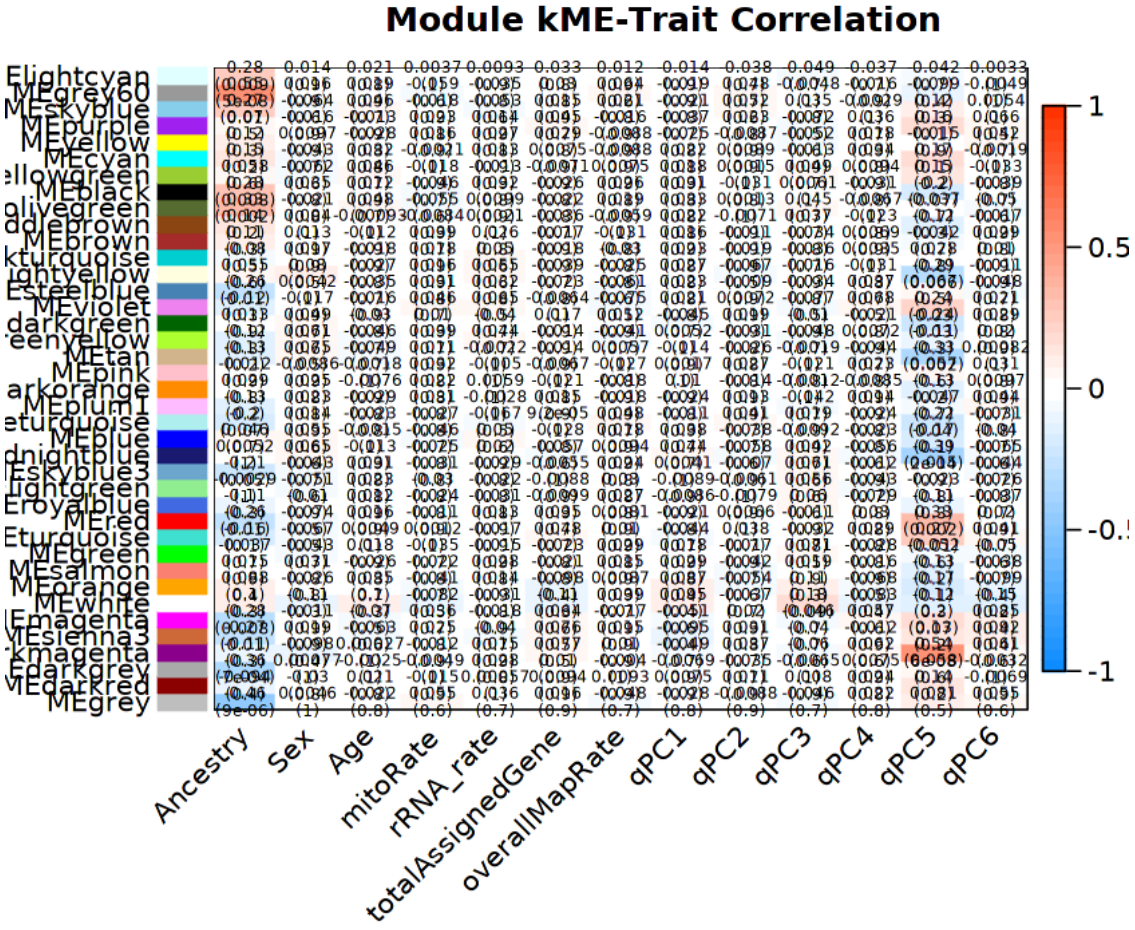
[14]:
```
#3 - TOM Dendogram
plot_cluster_dendrogram()
```

```
Allowing parallel execution with up to 16 working processes.
```



[15]:
```
#4 - Module Eigenvalue Correlation with sample's traits
correlate_with_traits()
```

```
Allowing parallel execution with up to 16 working processes.
```



[16]:
```
export_eigengene_tables()
```

1.9 Repreducibility Information

[17]:
```
Sys.time()
proc.time()
options(width = 120)
sessioninfo::session_info()
```

```
[1] "2021-07-12 10:15:15 EDT"
```

```
    user   system  elapsed
2607.378  841.370  550.362
```

```
 Session info
setting  value
version  R version 4.0.3 (2020-10-10)
os       Arch Linux
system   x86_64, linux-gnu
ui       X11
language (EN)
collate  en_US.UTF-8
ctype    en_US.UTF-8
tz       America/New_York
date     2021-07-12

 Packages
package        * version  date       lib source
AnnotationDbi    1.52.0   2020-10-27 [1] Bioconductor
assertthat       0.2.1    2019-03-21 [1] CRAN (R 4.0.2)
backports        1.2.1    2020-12-09 [1] CRAN (R 4.0.2)
base64enc        0.1-3    2015-07-28 [1] CRAN (R 4.0.2)
Biobase          2.50.0   2020-10-27 [1] Bioconductor
BiocGenerics     0.36.1   2021-04-16 [1] Bioconductor
bit              4.0.4    2020-08-04 [1] CRAN (R 4.0.2)
bit64            4.0.5    2020-08-30 [1] CRAN (R 4.0.2)
blob             1.2.1    2020-01-20 [1] CRAN (R 4.0.2)
cachem           1.0.5    2021-05-15 [1] CRAN (R 4.0.3)
Cairo            1.5-12.2 2020-07-07 [1] CRAN (R 4.0.2)
checkmate        2.0.0    2020-02-06 [1] CRAN (R 4.0.2)
cli              3.0.0    2021-06-30 [1] CRAN (R 4.0.3)
cluster          2.1.0    2019-06-19 [2] CRAN (R 4.0.3)
codetools        0.2-16   2018-12-24 [2] CRAN (R 4.0.3)
colorspace       2.0-2    2021-06-24 [1] CRAN (R 4.0.3)
crayon           1.4.1    2021-02-08 [1] CRAN (R 4.0.3)
data.table       1.14.0   2021-02-21 [1] CRAN (R 4.0.3)
DBI              1.1.1    2021-01-15 [1] CRAN (R 4.0.2)
digest           0.6.27   2020-10-24 [1] CRAN (R 4.0.2)
doParallel       1.0.16   2020-10-16 [1] CRAN (R 4.0.3)
dplyr          * 1.0.7    2021-06-18 [1] CRAN (R 4.0.3)
dynamicTreeCut * 1.63-1   2016-03-11 [1] CRAN (R 4.0.3)
ellipsis         0.3.2    2021-04-29 [1] CRAN (R 4.0.3)
evaluate         0.14     2019-05-28 [1] CRAN (R 4.0.2)
fansi            0.5.0    2021-05-25 [1] CRAN (R 4.0.3)
fastcluster    * 1.2.3    2021-05-24 [1] CRAN (R 4.0.3)
fastmap          1.1.0    2021-01-25 [1] CRAN (R 4.0.2)
foreach          1.5.1    2020-10-15 [1] CRAN (R 4.0.2)
foreign          0.8-80   2020-05-24 [2] CRAN (R 4.0.3)
Formula          1.2-4    2020-10-16 [1] CRAN (R 4.0.2)
generics         0.1.0    2020-10-31 [1] CRAN (R 4.0.2)
ggplot2          3.3.5    2021-06-25 [1] CRAN (R 4.0.3)
glue             1.4.2    2020-08-27 [1] CRAN (R 4.0.2)
```

```
GO.db            3.12.1   2021-04-08 [1] Bioconductor
gridExtra        2.3      2017-09-09 [1] CRAN (R 4.0.2)
gtable           0.3.0    2019-03-25 [1] CRAN (R 4.0.2)
Hmisc            4.5-0    2021-02-28 [1] CRAN (R 4.0.3)
htmlTable        2.2.1    2021-05-18 [1] CRAN (R 4.0.3)
htmltools        0.5.1.1  2021-01-22 [1] CRAN (R 4.0.2)
htmlwidgets      1.5.3    2020-12-10 [1] CRAN (R 4.0.2)
impute           1.64.0   2020-10-27 [1] Bioconductor
IRanges          2.24.1   2020-12-12 [1] Bioconductor
IRdisplay        1.0      2021-01-20 [1] CRAN (R 4.0.2)
IRkernel         1.2      2021-05-11 [1] CRAN (R 4.0.3)
iterators        1.0.13   2020-10-15 [1] CRAN (R 4.0.2)
jpeg             0.1-8.1  2019-10-24 [1] CRAN (R 4.0.2)
jsonlite         1.7.2    2020-12-09 [1] CRAN (R 4.0.2)
knitr            1.33     2021-04-24 [1] CRAN (R 4.0.3)
lattice          0.20-41  2020-04-02 [2] CRAN (R 4.0.3)
latticeExtra     0.6-29   2019-12-19 [1] CRAN (R 4.0.2)
lifecycle        1.0.0    2021-02-15 [1] CRAN (R 4.0.3)
limma          * 3.46.0   2020-10-27 [1] Bioconductor
magrittr         2.0.1    2020-11-17 [1] CRAN (R 4.0.2)
Matrix           1.3-4    2021-06-01 [1] CRAN (R 4.0.3)
matrixStats      0.59.0   2021-06-01 [1] CRAN (R 4.0.3)
memoise          2.0.0    2021-01-26 [1] CRAN (R 4.0.2)
munsell          0.5.0    2018-06-12 [1] CRAN (R 4.0.2)
nnet             7.3-14   2020-04-26 [2] CRAN (R 4.0.3)
pbdZMQ           0.3-5    2021-02-10 [1] CRAN (R 4.0.3)
pillar           1.6.1    2021-05-16 [1] CRAN (R 4.0.3)
pkgconfig        2.0.3    2019-09-22 [1] CRAN (R 4.0.2)
png              0.1-7    2013-12-03 [1] CRAN (R 4.0.2)
preprocessCore   1.52.1   2021-01-08 [1] Bioconductor
purrr            0.3.4    2020-04-17 [1] CRAN (R 4.0.2)
R6               2.5.0    2020-10-28 [1] CRAN (R 4.0.2)
RColorBrewer     1.1-2    2014-12-07 [1] CRAN (R 4.0.2)
Rcpp             1.0.7    2021-07-07 [1] CRAN (R 4.0.3)
repr             1.1.3    2021-01-21 [1] CRAN (R 4.0.2)
rlang            0.4.11   2021-04-30 [1] CRAN (R 4.0.3)
rpart            4.1-15   2019-04-12 [2] CRAN (R 4.0.3)
RSQLite          2.2.7    2021-04-22 [1] CRAN (R 4.0.3)
rstudioapi       0.13     2020-11-12 [1] CRAN (R 4.0.2)
S4Vectors        0.28.1   2020-12-09 [1] Bioconductor
scales           1.1.1    2020-05-11 [1] CRAN (R 4.0.2)
sessioninfo      1.1.1    2018-11-05 [1] CRAN (R 4.0.2)
stringi          1.6.2    2021-05-17 [1] CRAN (R 4.0.3)
stringr          1.4.0    2019-02-10 [1] CRAN (R 4.0.2)
survival         3.2-7    2020-09-28 [2] CRAN (R 4.0.3)
tibble           3.1.2    2021-05-16 [1] CRAN (R 4.0.3)
tidyselect       1.1.1    2021-04-30 [1] CRAN (R 4.0.3)
utf8             1.2.1    2021-03-12 [1] CRAN (R 4.0.3)
```

```
 uuid             0.1-4    2020-02-26 [1] CRAN (R 4.0.2)
 vctrs            0.3.8    2021-04-29 [1] CRAN (R 4.0.3)
 WGCNA          * 1.70-3   2021-02-28 [1] CRAN (R 4.0.3)
 withr            2.4.2    2021-04-18 [1] CRAN (R 4.0.3)
 xfun             0.24     2021-06-15 [1] CRAN (R 4.0.3)

[1] /home/jbenja13/R/x86_64-pc-linux-gnu-library/4.0
[2] /usr/lib/R/library
```