Metagenomic_analysis

Sergio Rodriguez Llana

7/20/2020

Bray-Curtis dissimilarity matrix analysis (SIMKA)

Analysis done to the Bray-Curtis dissimilarity matrix resulting using the program **SIMKA**.

Loading Vegan package:

```
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.5-6
```

Loading the abundance Bray-Curtis matrix. **Note**: The sample names have been changed from the original in order to interpret them better.

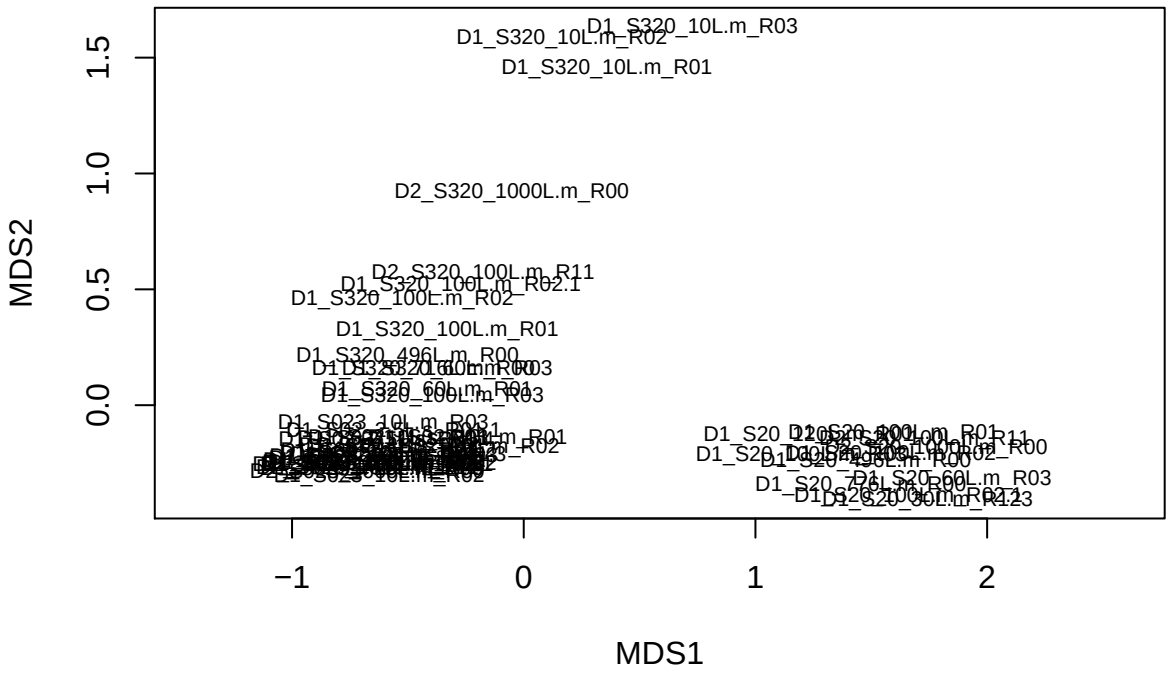
I loaded the matrix twice to set the matrix correctly with the appropiate row names corresponding to the samples (otherwise I get an error).

NMDS

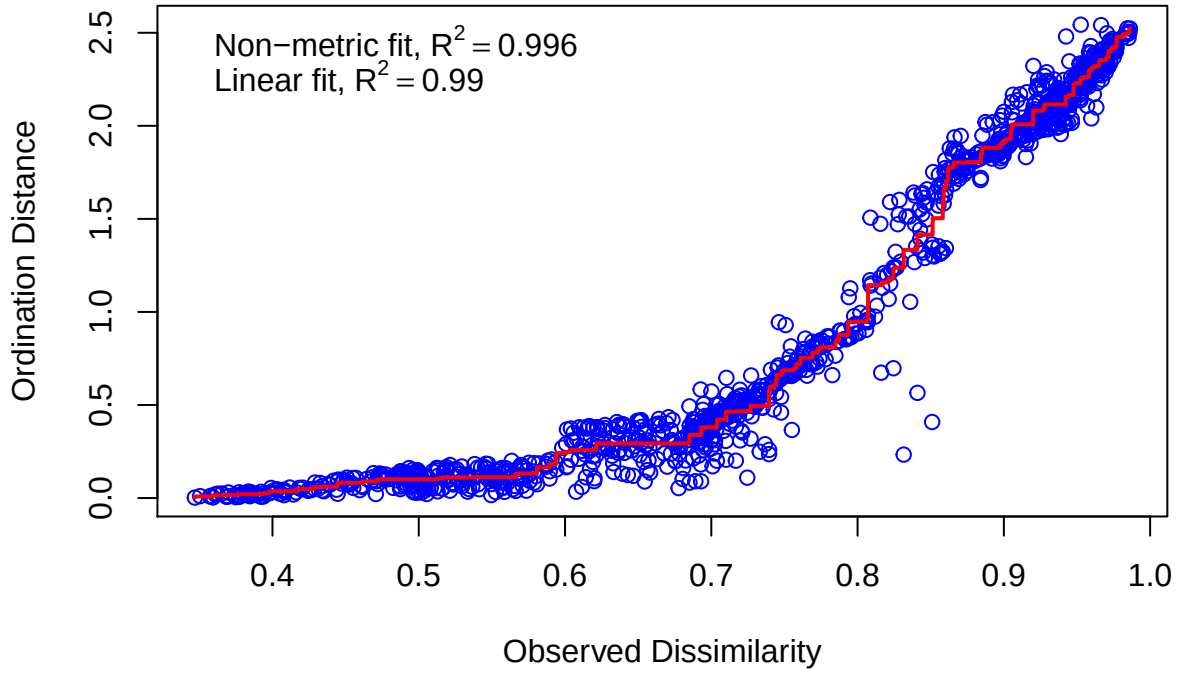
```
simka_nmds <- monoMDS(simka_tab)
simka_nmds
```

```
##
## Call:
## monoMDS(dist = simka_tab)
##
## Non-metric Multidimensional Scaling
##
## 50 points, dissimilarity 'unknown'
##
## Dimensions: 2
## Stress:     0.06488936
## Stress type 1, weak ties
## Scores scaled to unit root mean square, rotated to principal components
## Stopped after 83 iterations: Stress nearly unchanged (ratio > sratmax)
```

```
plot(simka_nmds)
```



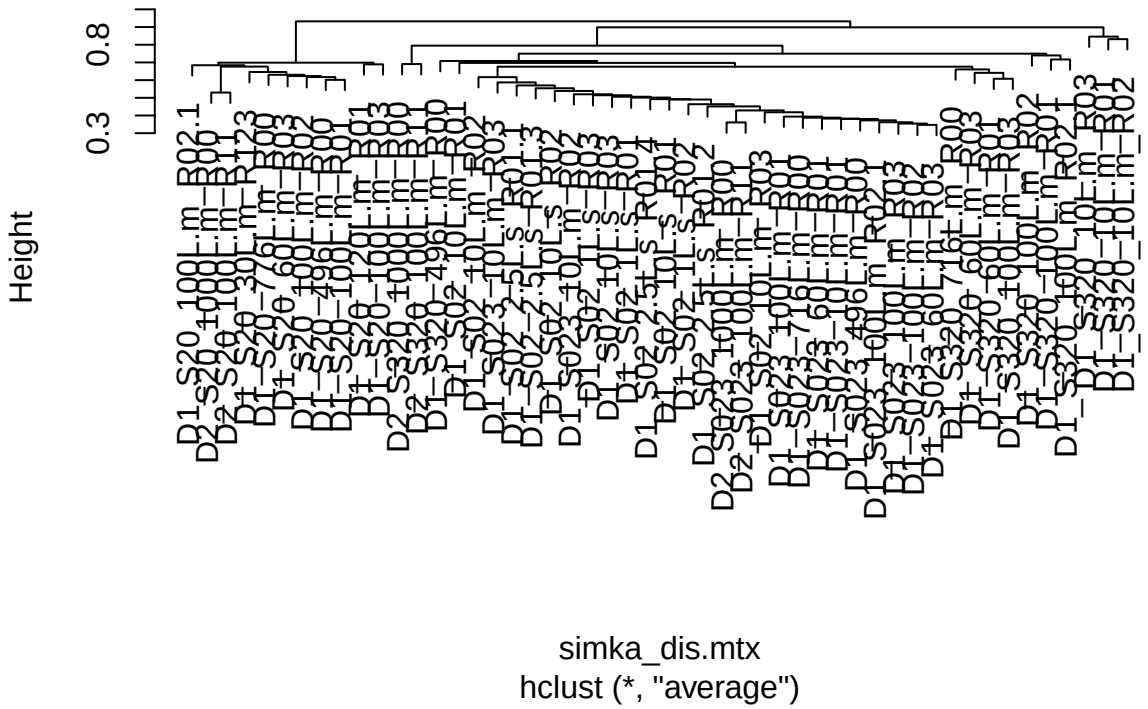

```
stressplot(simka_nmds)
```



UPGMA

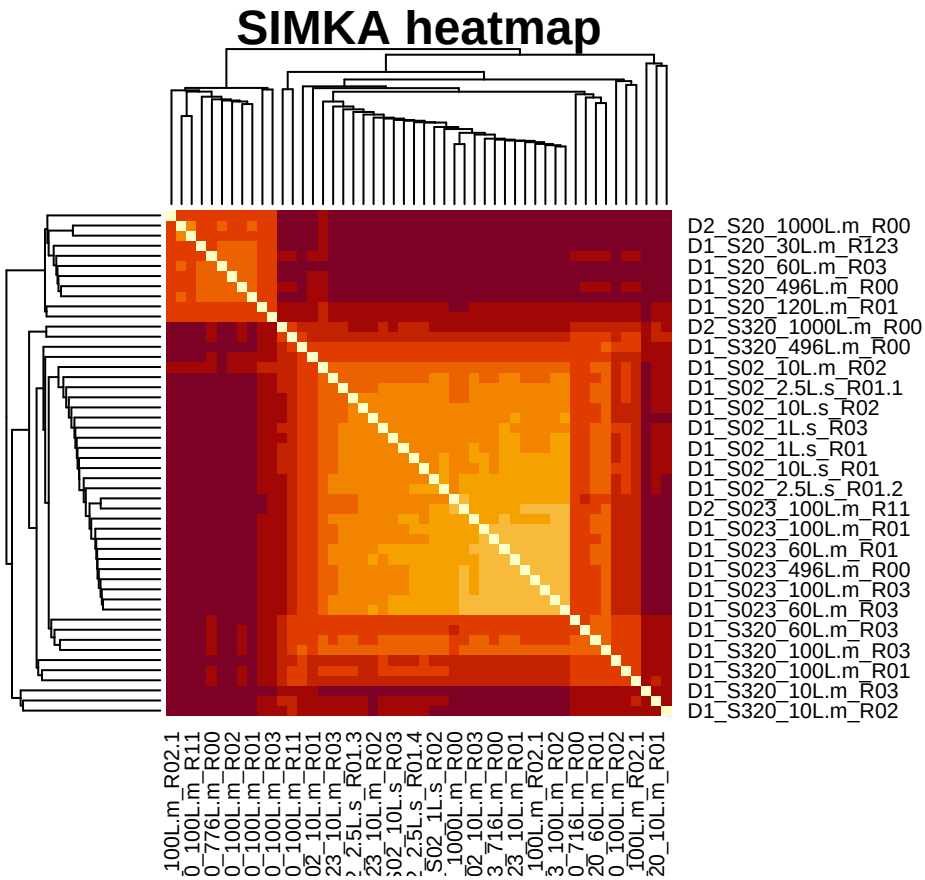
Dendogram generated using UPGMA clustering algorithm:

```
simka_dis.mtx <- as.dist(simka_tab)
simka_hclust <- hclust(simka_dis.mtx, method = "average")
plot(simka_hclust, main = "SIMKA dendogram")
```



Heatmap

```
simka_mtx <- as.matrix(simka_tab)
simka_dendo <- as.dendrogram(simka_hclust)
heatmap(simka_mtx, Rowv = simka_dendo, Colv = "Rowv", symm = TRUE, main = "SIMKA heatmap")
```



Function abundance tables

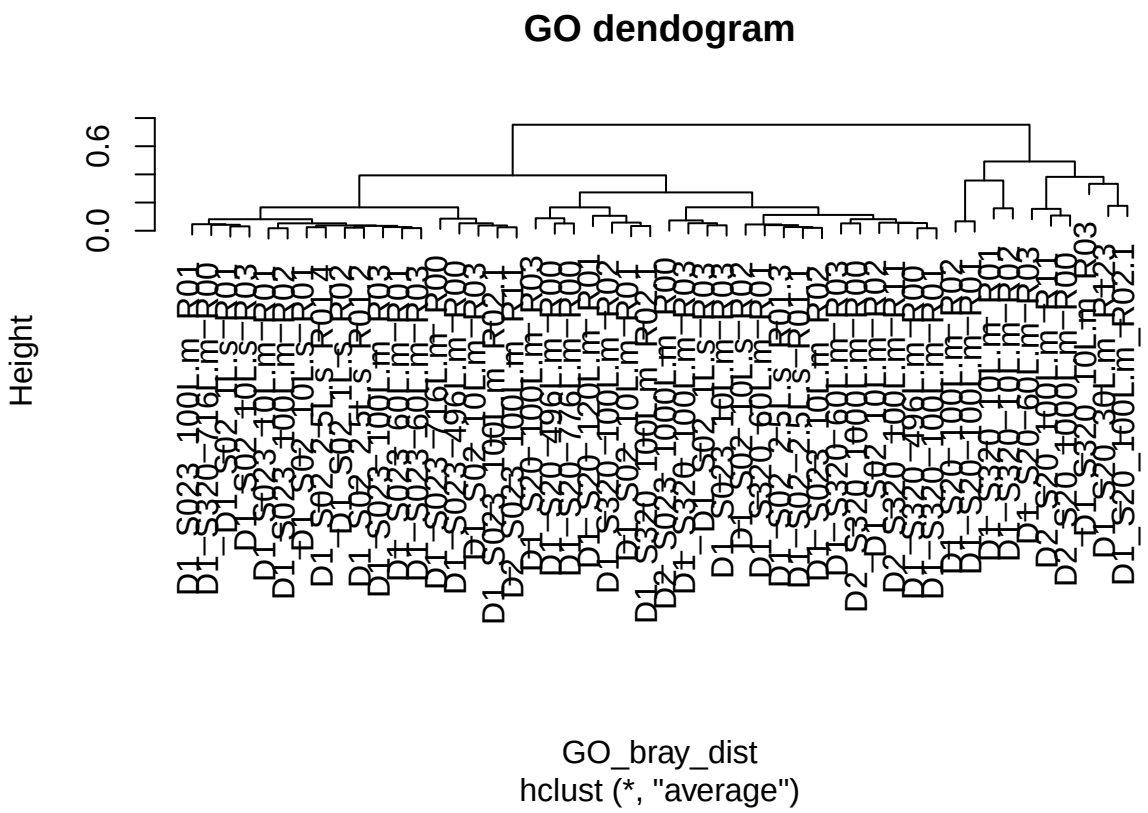
Computation for Bray-Curtis dissimilarity matrices for GO, GO-slim and Interpro (IPR) abundance tables.

GO dissimilarity matrix

```
GO_table <- read.table(file = "/home/sergio/Desktop/Internship/R/ERP112966_GO_abundances_v4.1.tsv",
                       header = TRUE, sep = "\t", row.names = 1)
GO_table$description <- NULL
GO_table$category <- NULL
GO_ttable <- t(GO_table)
GO_ttable <- as.data.frame(GO_ttable)
#head(GO_ttable, 10)
GO_ttable <- as.matrix(GO_ttable)

#Generating the dissimilarity matrix
GO_bray_dist <- vegdist(GO_ttable, method = "bray")
#head(as.matrix(GO_bray_dist), 5)
```

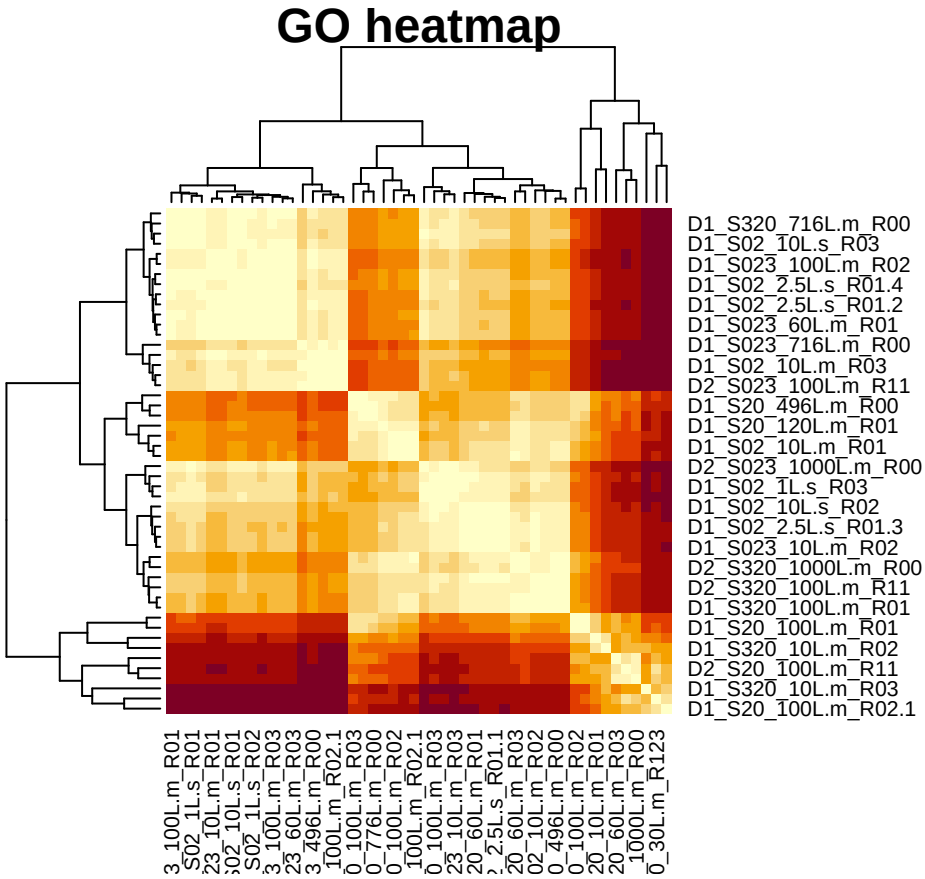
GO UPGMA

```
GO_bray_hclust <- hclust(GO_bray_dist, method = "average")
plot(GO_bray_hclust, main = "GO dendogram")
```



GO Heatmap

```
GO_bray_mtx <- as.matrix(GO_bray_dist)
GO_bray_dendo <- as.dendrogram(GO_bray_hclust)
heatmap(GO_bray_mtx, Rowv = GO_bray_dendo, Colv = "Rowv", symm = TRUE, main = "GO heatmap")
```

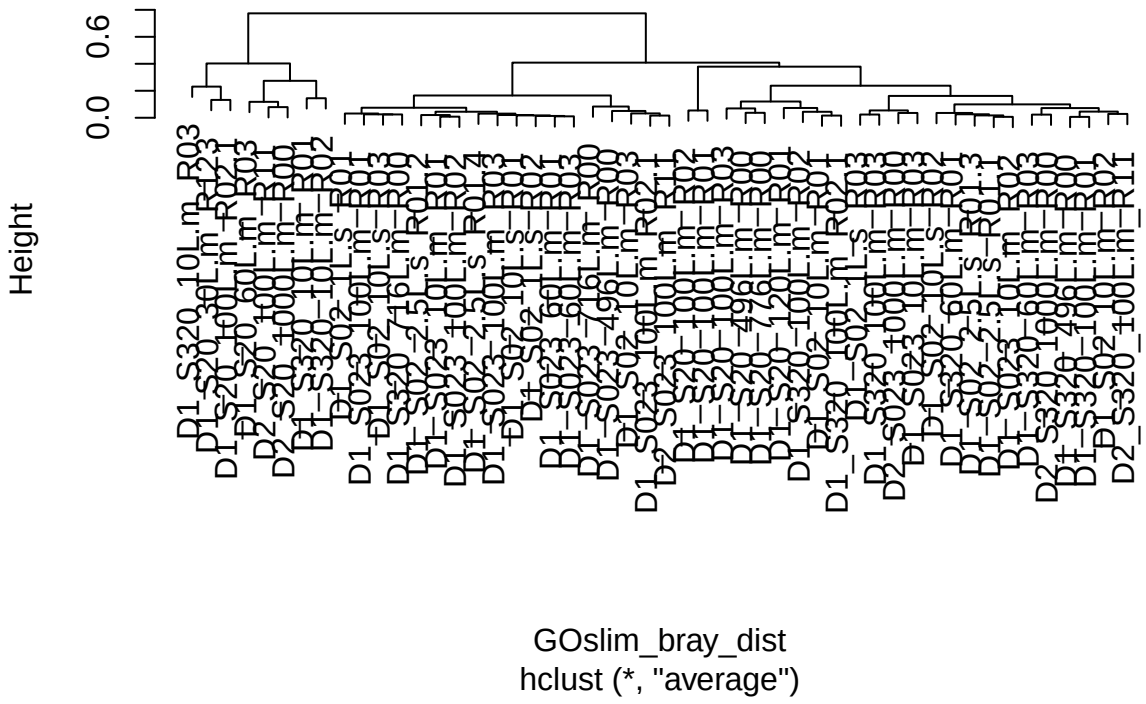


GO-slim dissimilarity matrix

```
GOslim_table <- read.table(file = "/home/sergio/Desktop/Internship/R/ERP112966_GO-slim_abundances_v4.1.t
                      header = TRUE, sep = "\t", row.names = 1)
GOslim_table$description <- NULL
GOslim_table$category <- NULL
GOslim_ttable <- t(GOslim_table)
GOslim_ttable <- as.data.frame(GOslim_ttable)
#head(GOslim_ttable, 10)
GOslim_ttable <- as.matrix(GOslim_ttable)

#Generating the dissimilarity matrix
GOslim_bray_dist <- vegdist(GOslim_ttable, method = "bray")
#head(as.matrix(GOslim_bray_dist), 5)
```

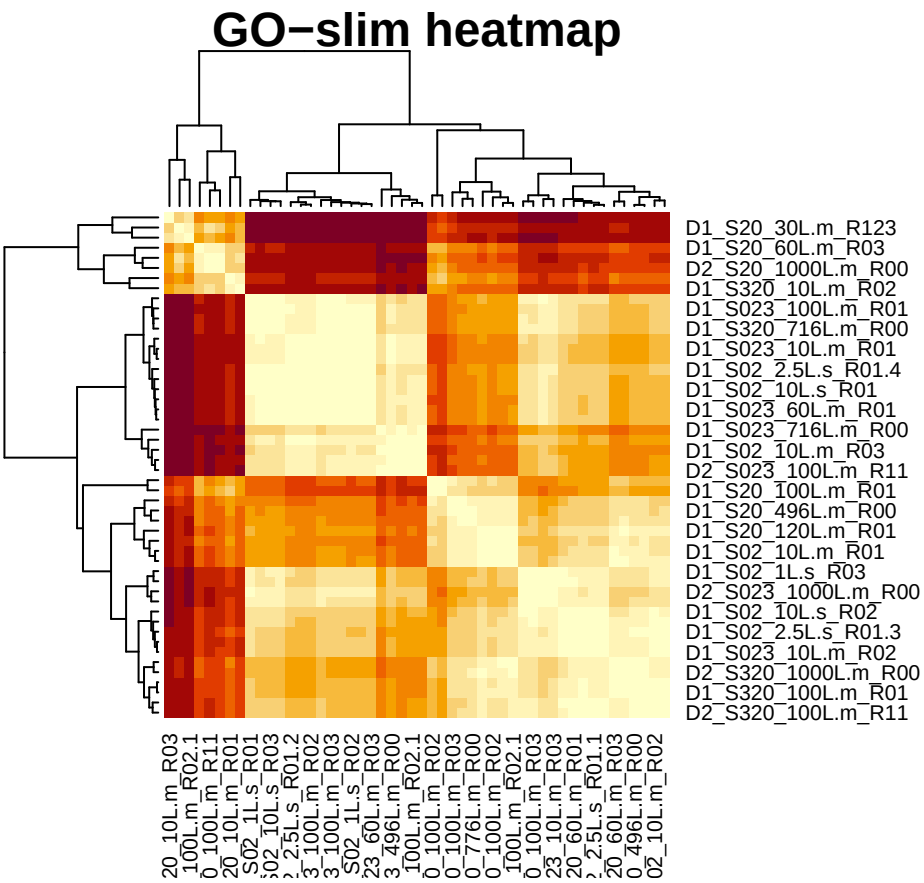
GO-slim UPGMA

```
GOslim_bray_hclust <- hclust(GOslim_bray_dist, method = "average")
plot(GOslim_bray_hclust, main = "GO-slim dendogram")
```



GO-slim heatmap

```
GOslim_bray_mtx <- as.matrix(GOslim_bray_dist)
GOslim_bray_dendo <- as.dendrogram(GOslim_bray_hclust)
heatmap(GOslim_bray_mtx, Rowv = GOslim_bray_dendo, Colv = "Rowv", symm = TRUE, main = "GO-slim heatmap")
```

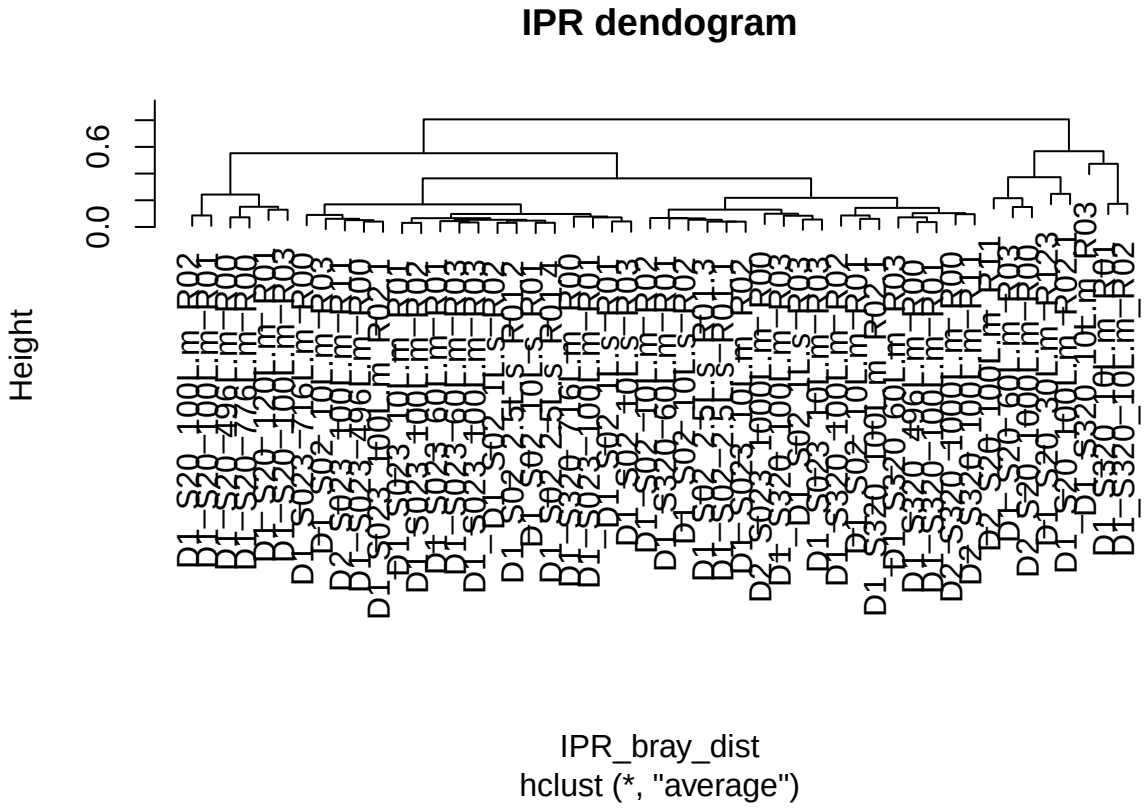


Interpro dissimilarity matrix

```
IPR_table <- read.table(file = "/home/sergio/Desktop/Internship/R/ERP112966_IPR_abundances_v4.1.tsv",
                        header = TRUE, sep = "\t", row.names = 1)
IPR_table$description <- NULL
IPR_ttable <- t(IPR_table)
IPR_ttable <- as.data.frame(IPR_ttable)
#head(IPR_ttable, 10)
IPR_ttable <- as.matrix(IPR_ttable)

#Generating the dissimilarity matrix
IPR_bray_dist <- vegdist(IPR_ttable, method = "bray")
#head(as.matrix(IPR_bray_dist), 5)
```

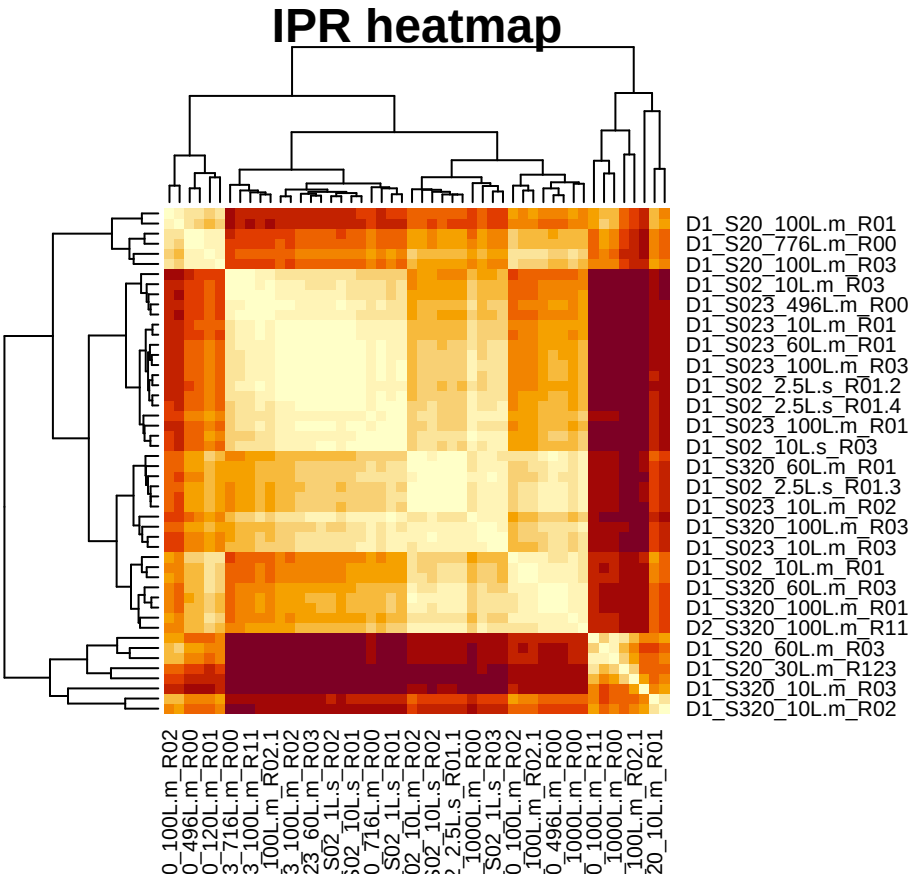
Interpro UPGMA

```
IPR_bray_hclust <- hclust(IPR_bray_dist, method = "average")
plot(IPR_bray_hclust, main = "IPR dendogram")
```



Interpro heatmap

```
IPR_bray_mtx <- as.matrix(IPR_bray_dist)
IPR_bray_dendo <- as.dendrogram(IPR_bray_hclust)
heatmap(IPR_bray_mtx, Rowv = IPR_bray_dendo, Colv = "Rowv", symm = TRUE, main = "IPR heatmap")
```



Analysis of Bray-Curtis matrices for the GO abundace table with and without subsampling

Bray-Curtis matrices for the GO table table done without subsampling and subsampling:

- No subsampling (NS)
- Subsampling: Minimum sum of row values (SS_MIN)
- Subsampling: Mean sum of row values (SS_MEAN)

The NS dissimilarity matrix is already generated at this point ('GO_bray_dist').

Generating SS_MIN and SS_MEAN distance dissimilarity matrices

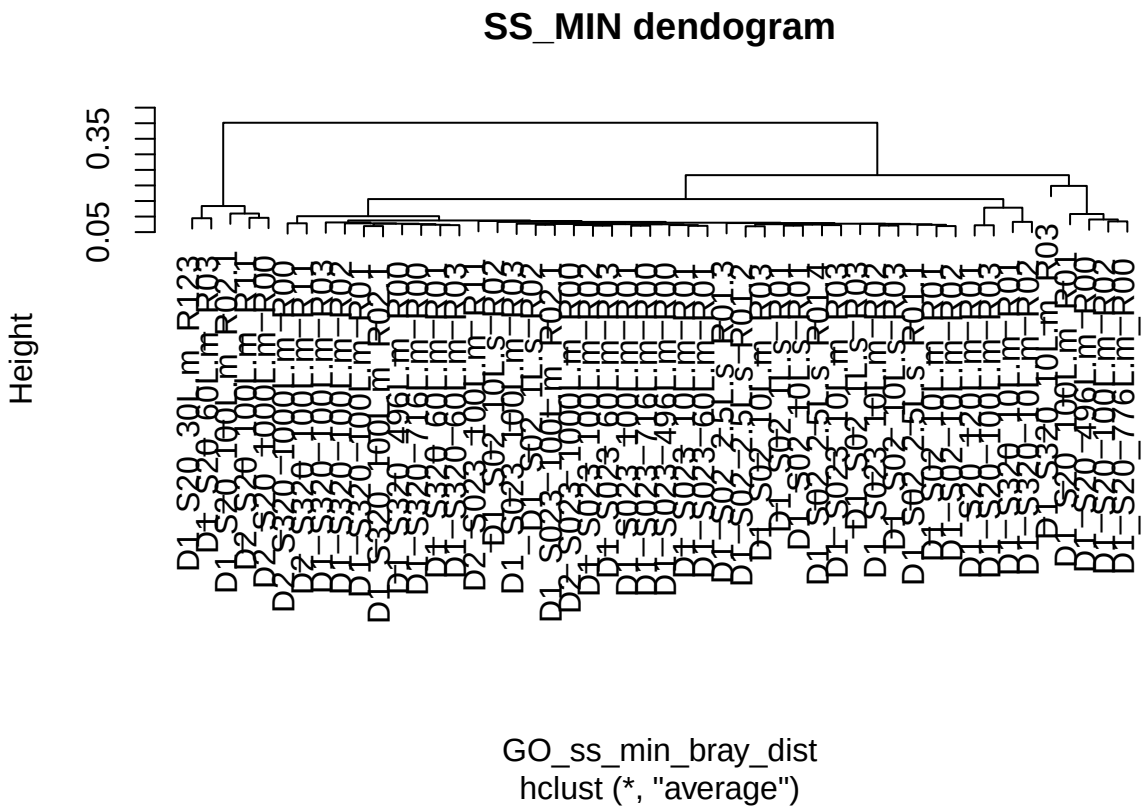
In addition to the distance matrices, the dendograms (UPGMA) and heatmaps are also included.

```
print(paste("Minimum number of reads =", min(rowSums(GO_ttable))))
```

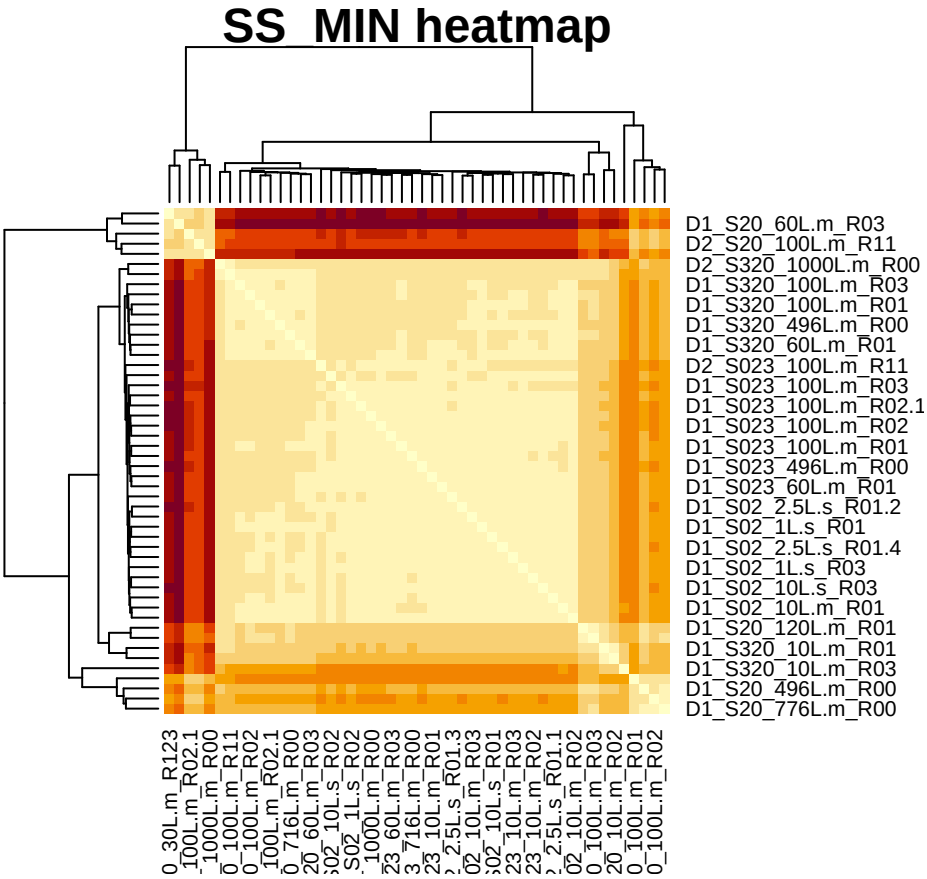
```
## [1] "Minimum number of reads = 22506"
```

```
GO_table_ss_min <- rrarefy(GO_ttable, 22506)

#Obtain a Bray Curtis dissimilarity matrix:
GO_ss_min_bray_dist <- vegdist(GO_table_ss_min, method = "bray")

#hclust UPGMA
```

```
GO_ss_min_hclust <- hclust(GO_ss_min_bray_dist, method = "average")
plot(GO_ss_min_hclust, main = "SS_MIN dendogram")
```




```
#Heatmap
GO_ss_min_bray_mtx <- as.matrix(GO_ss_min_bray_dist)
GO_ss_min_dendo <- as.dendrogram(GO_ss_min_hclust)
heatmap(GO_ss_min_bray_mtx, Rowv = GO_ss_min_dendo, Colv = "Rowv", symm = TRUE, main = "SS_MIN heatmap")
```



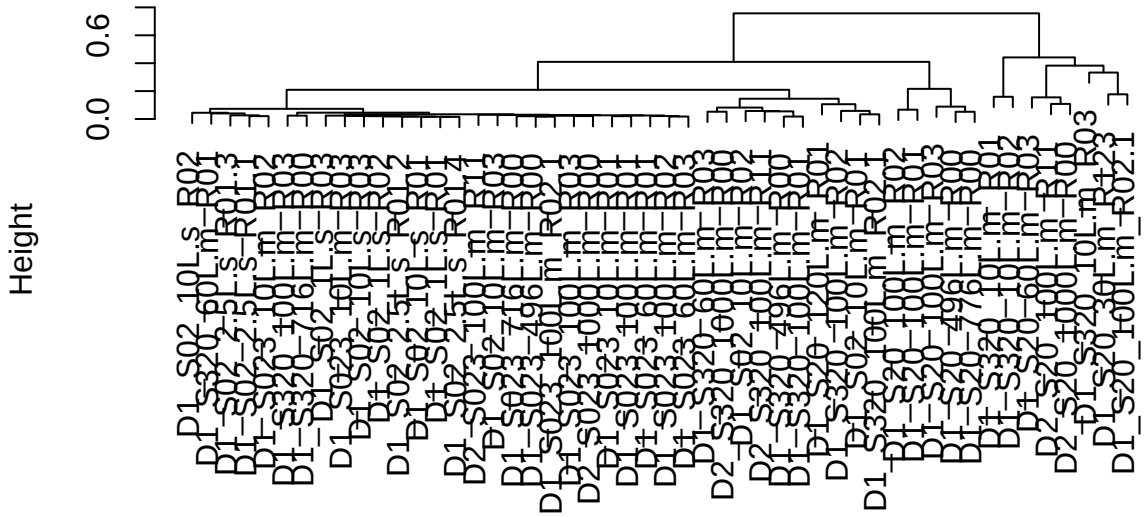

```
print(paste("Mean number of reads =", mean(rowSums(GO_ttable))))
```

```
## [1] "Mean number of reads = 361134.78"
```

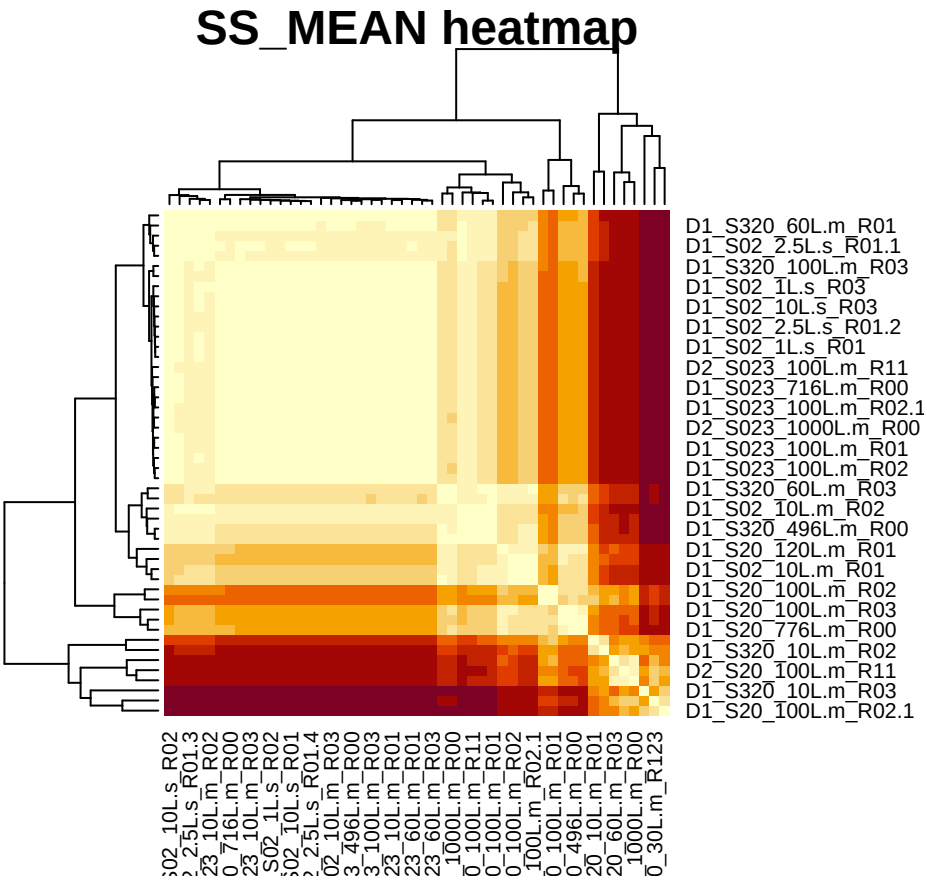
```
GO_table_ss_mean <- rrarefy(GO_ttable, 361135)
```

```
## Warning in rrarefy(GO_ttable, 361135): some row sums < 'sample' and are not
## rarefied
```

```
#Obtain a Bray Curtis dissimilarity matrix:
GO_ss_mean_bray_dist <- vegdist(GO_table_ss_mean, method = "bray")

#hclust UPGMA
GO_ss_mean_hclust <- hclust(GO_ss_mean_bray_dist, method = "average")
plot(GO_ss_mean_hclust, main = "SS_mean dendogram")
```

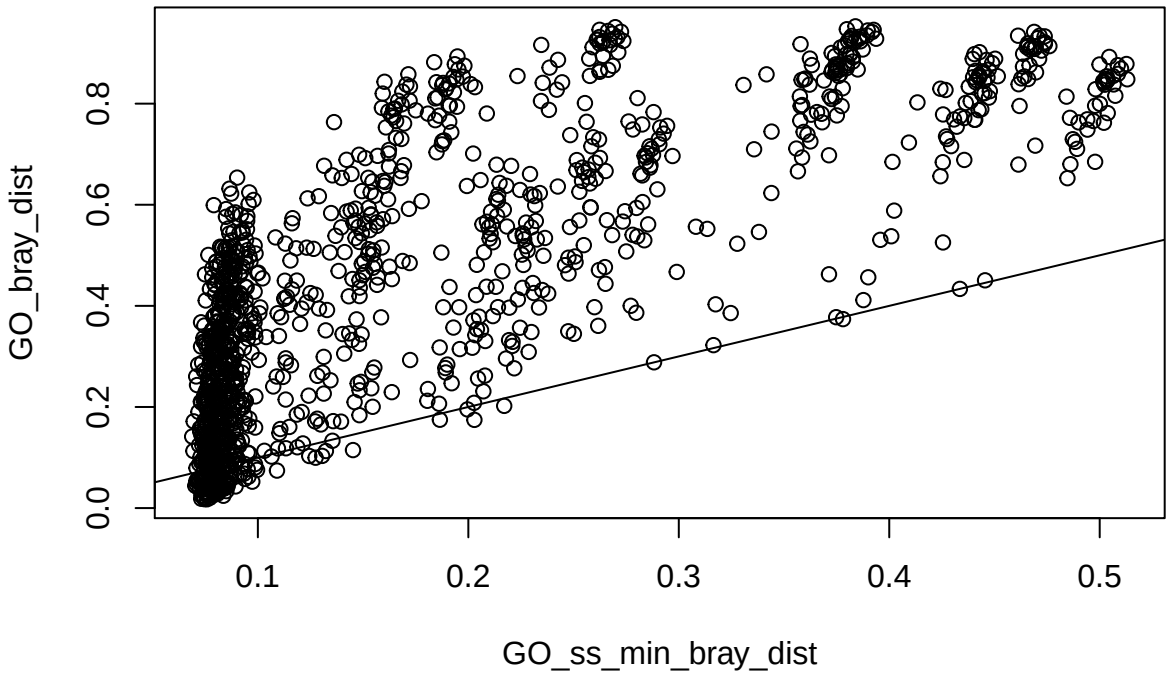



```
#Heatmap
GO_ss_mean_bray_mtx <- as.matrix(GO_ss_mean_bray_dist)
GO_ss_mean_dendo <- as.dendrogram(GO_ss_mean_hclust)
heatmap(GO_ss_mean_bray_mtx, Rowv = GO_ss_mean_dendo, Colv = "Rowv", symm = TRUE, main = "SS_MEAN heatma
```



Mantel Tests for dissimilarity matrices

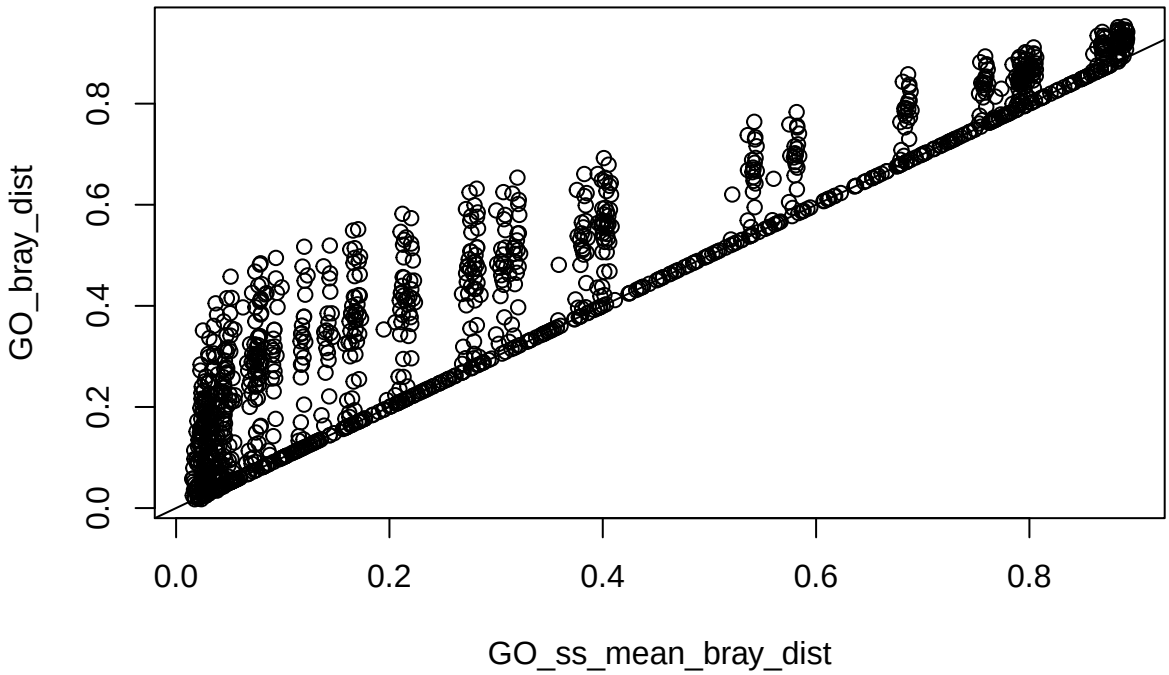
```
plot(GO_ss_min_bray_dist, GO_bray_dist, main = "SS_MIN vs NS")
abline(0,1)
```




```
mantel(GO_ss_min_bray_dist, GO_bray_dist)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = GO_ss_min_bray_dist, ydis = GO_bray_dist)
##
## Mantel statistic r: 0.7778
##       Significance: 0.001
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.117 0.165 0.201 0.251
## Permutation: free
## Number of permutations: 999
```

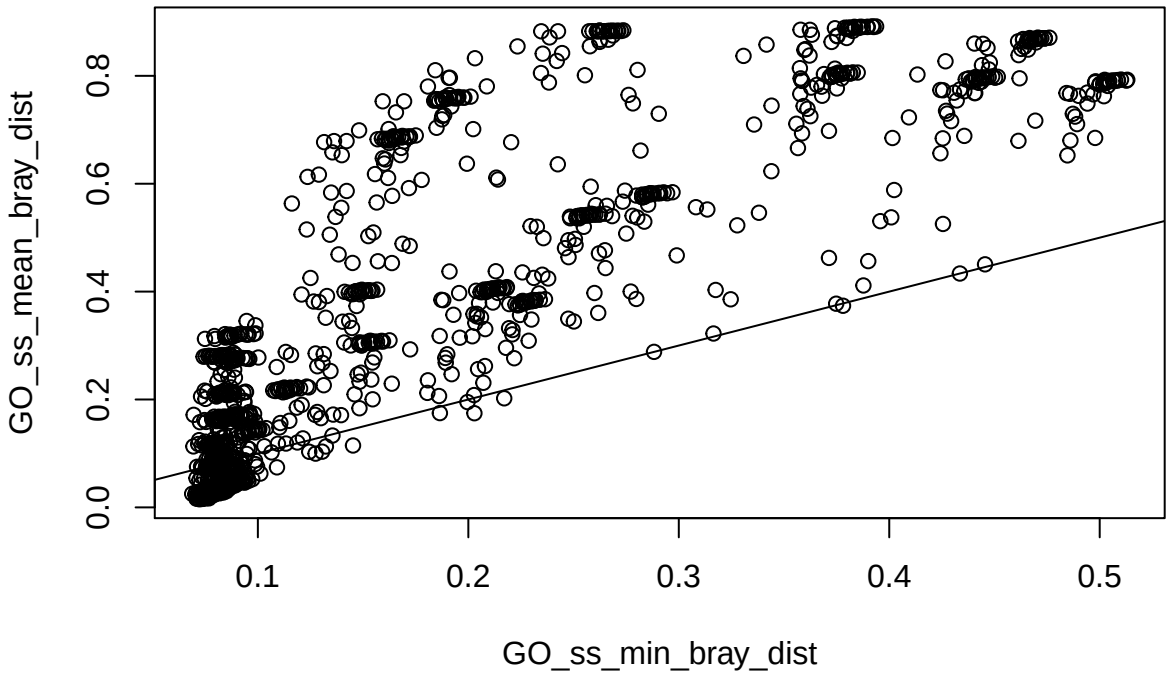
```
plot(GO_ss_mean_bray_dist, GO_bray_dist, main = "SS_MEAN vs NS")
abline(0,1)
```




```
mantel(GO_ss_mean_bray_dist, GO_bray_dist)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = GO_ss_mean_bray_dist, ydis = GO_bray_dist)
##
## Mantel statistic r: 0.9376
##       Significance: 0.001
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.104 0.142 0.186 0.244
## Permutation: free
## Number of permutations: 999
```

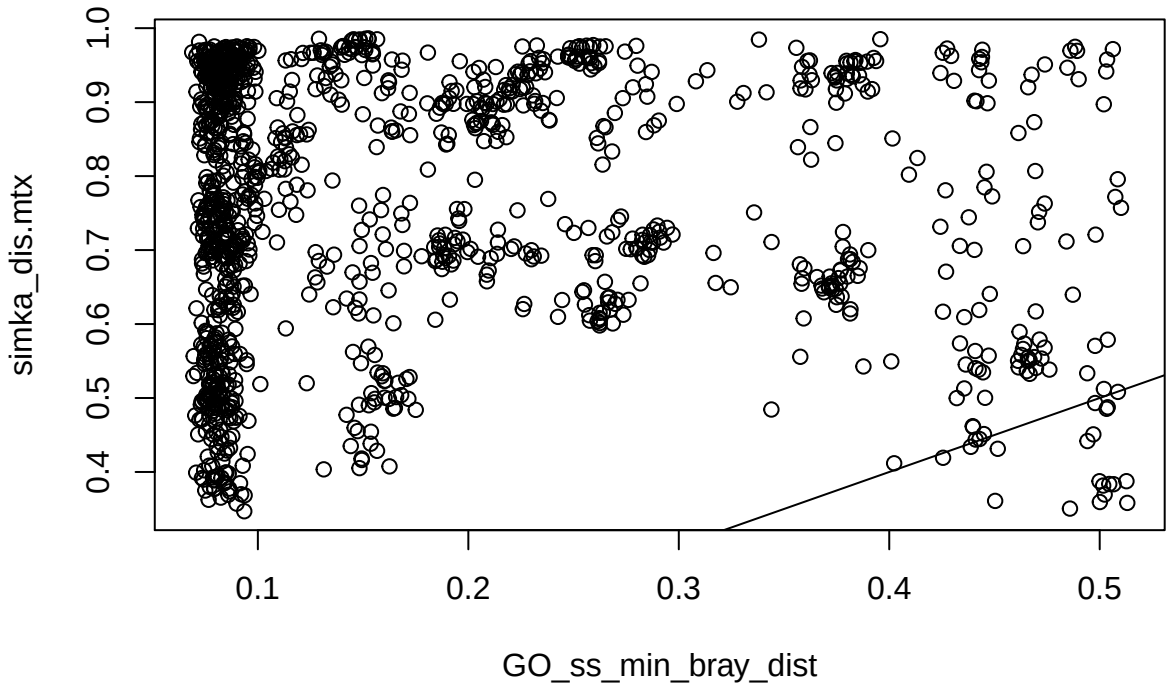
```
plot(GO_ss_min_bray_dist, GO_ss_mean_bray_dist, main = "SS_MIN vs SS_MEAN")
abline(0,1)
```




```
mantel(GO_ss_min_bray_dist, GO_ss_mean_bray_dist)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = GO_ss_min_bray_dist, ydis = GO_ss_mean_bray_dist)
##
## Mantel statistic r: 0.8545
##       Significance: 0.001
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.125 0.176 0.226 0.305
## Permutation: free
## Number of permutations: 999
```

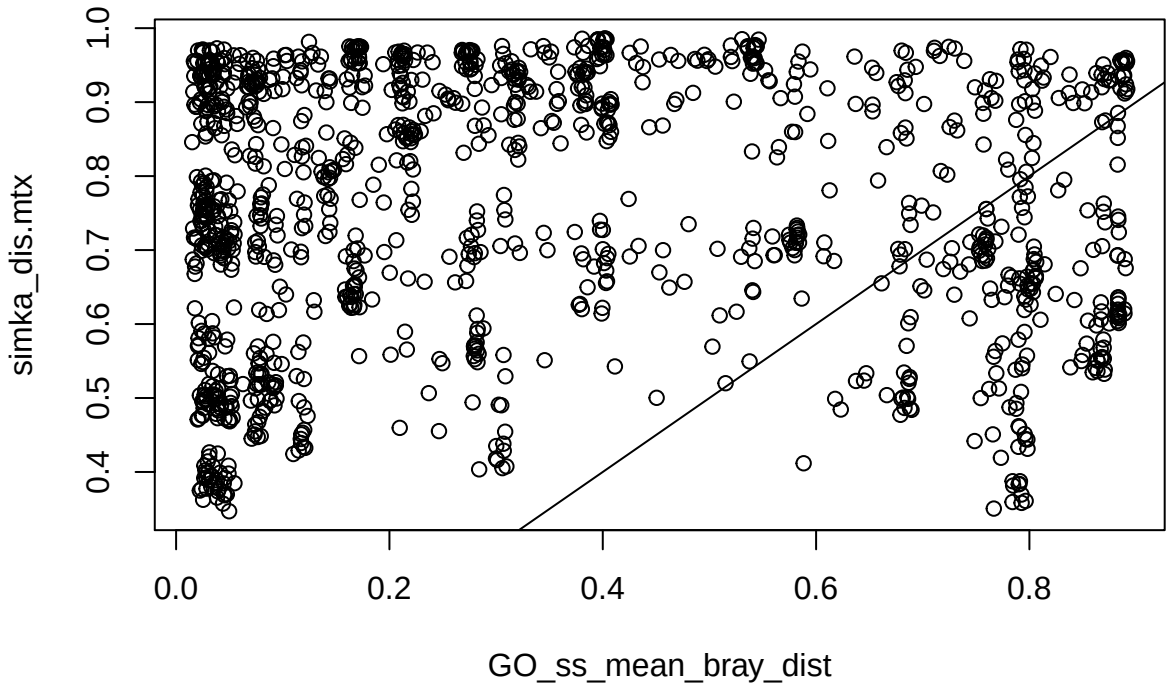
```
plot(GO_ss_min_bray_dist, simka_dis.mtx, main = "SS_MIN vs SIMKA")
abline(0,1)
```




```
mantel(GO_ss_min_bray_dist, simka_dis.mtx)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = GO_ss_min_bray_dist, ydis = simka_dis.mtx)
##
## Mantel statistic r: -0.04446
##       Significance: 0.646
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.122 0.161 0.197 0.239
## Permutation: free
## Number of permutations: 999
```

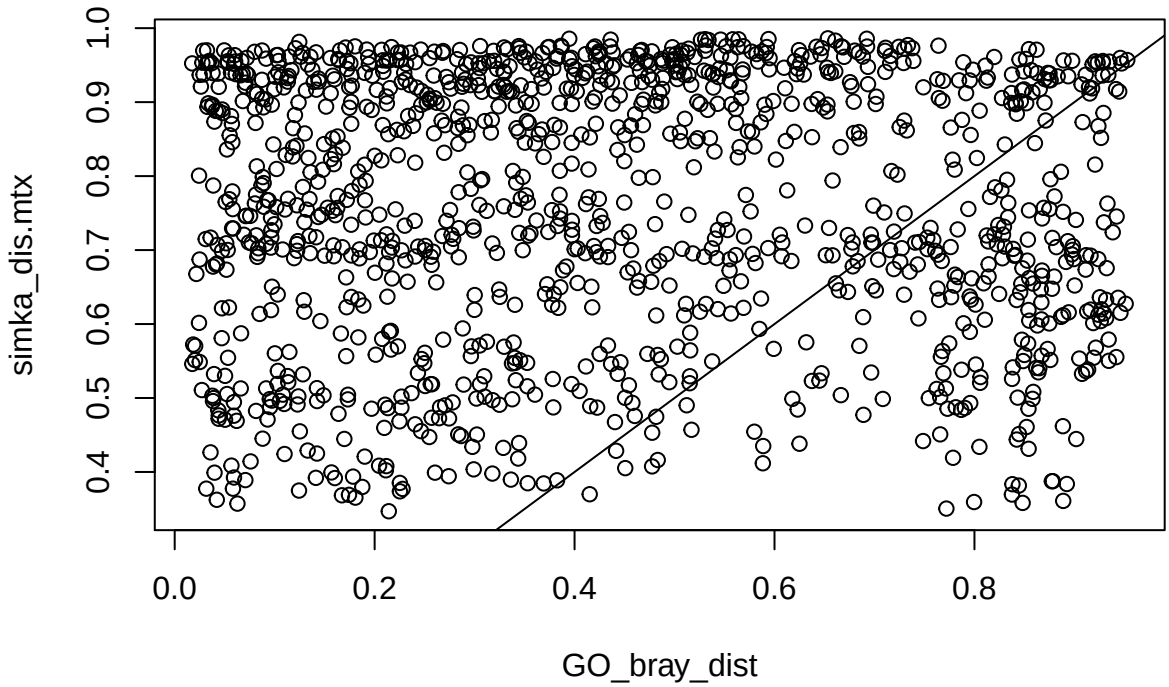
```
plot(GO_ss_mean_bray_dist, simka_dis.mtx, main = "SS_MEAN vs SIMKA")
abline(0,1)
```




```
mantel(GO_ss_mean_bray_dist, simka_dis.mtx)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = GO_ss_mean_bray_dist, ydis = simka_dis.mtx)
##
## Mantel statistic r: 0.01565
##       Significance: 0.361
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.108 0.143 0.166 0.210
## Permutation: free
## Number of permutations: 999
```

```
plot(GO_bray_dist, simka_dis.mtx, main = "NS vs SIMKA")
abline(0,1)
```




```
mantel(GO_bray_dist, simka_dis.mtx)
```

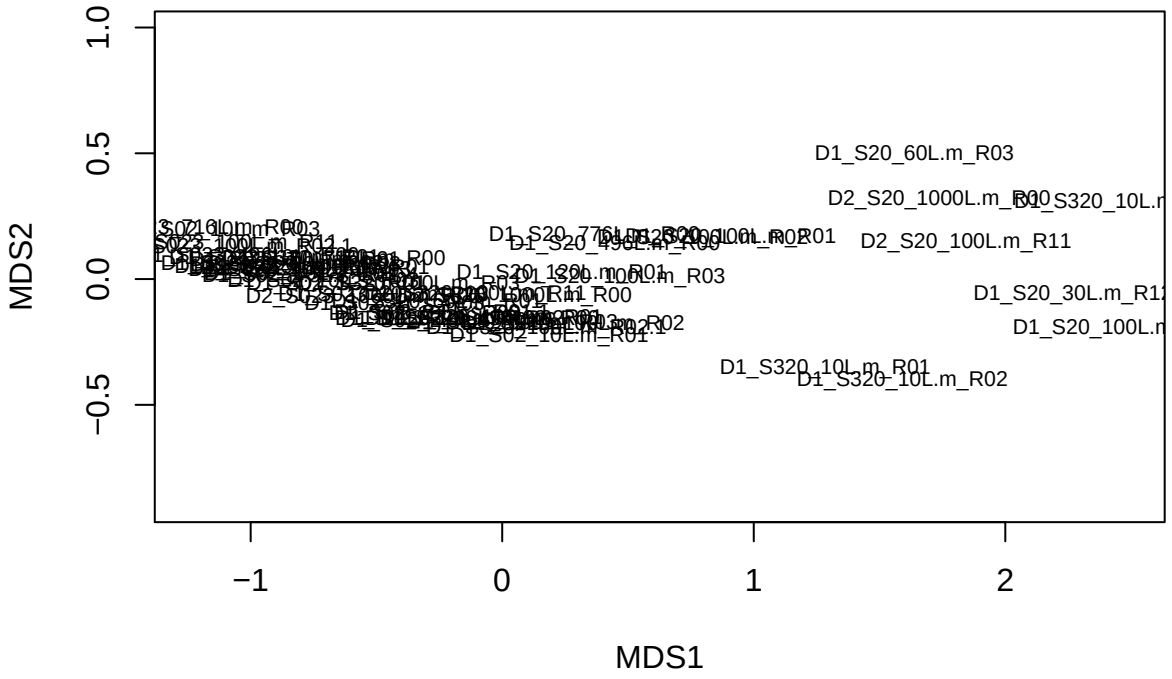
```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = GO_bray_dist, ydis = simka_dis.mtx)
##
## Mantel statistic r: -0.007522
##       Significance: 0.487
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.108 0.136 0.164 0.206
## Permutation: free
## Number of permutations: 999
```

Good correlation between the GO dissimilarity matrices, although there is a lot of dispersion in the plots Bad correation between SIMKA and GO dissimilarity matrices

NMDS

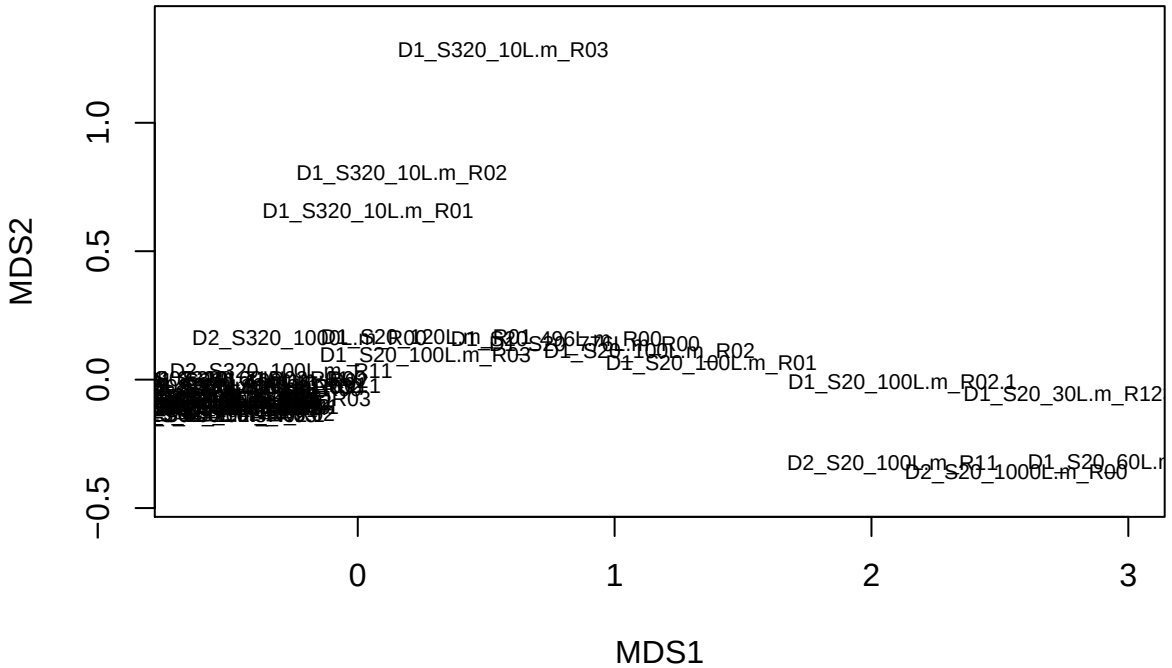
```
GO_ns_bray_nmds <- monoMDS(GO_bray_dist)
plot(GO_ns_bray_nmds, main = "NS NMDS")
```

NS NMDS

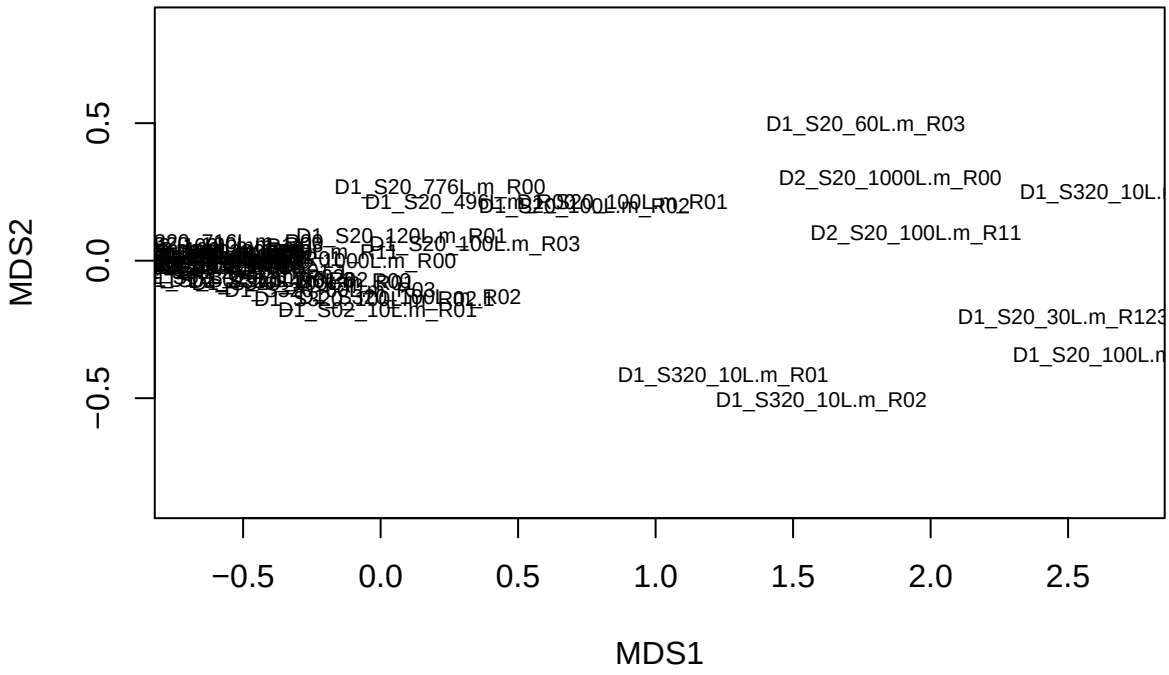



```
GO_ss_min_bray_nmds <- monoMDS(GO_ss_min_bray_dist)
plot(GO_ss_min_bray_nmds, main = "SS_MIN NMDS")
```

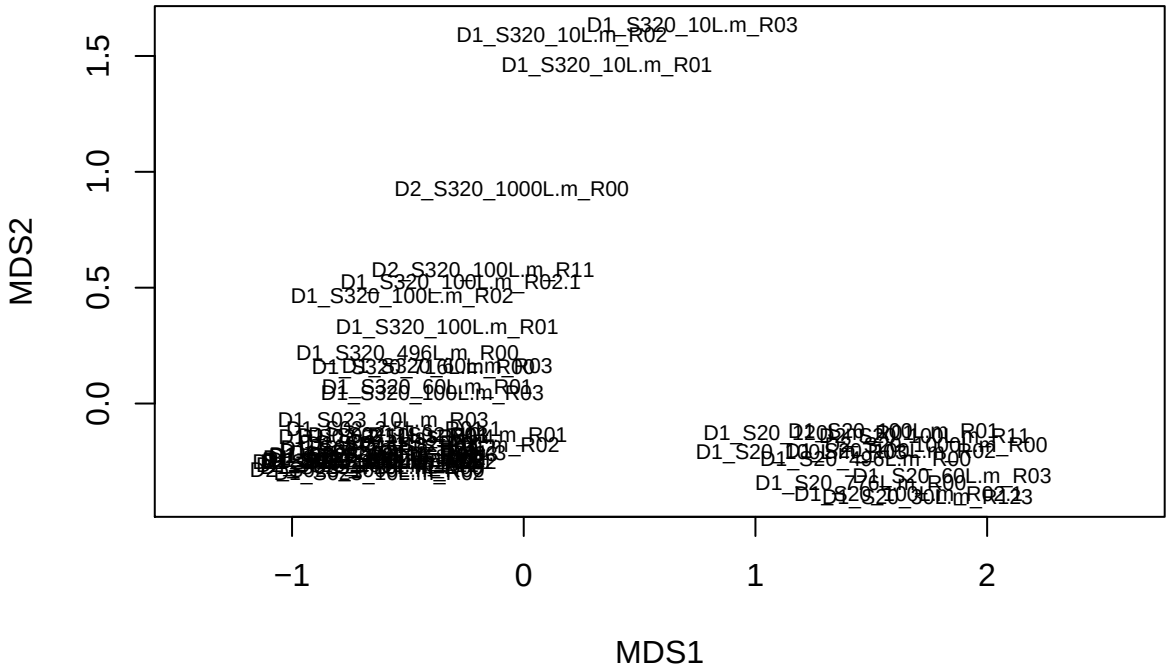
SS_MIN NMDS




```
GO_ss_mean_bray_nmds <- monoMDS(GO_ss_mean_bray_dist)
plot(GO_ss_mean_bray_nmds, main = "SS_MEAN NMDS")
```




```
plot(simka_nmds, main = "SIMKA NMDS")
```



We cannot form clear sample clusters of the GO dis. matrices

ANOSIM NS

Function to reorder the rows and columns of a symmetric matrix (from package 'graph4lg'):

```
reorder_mat <- function(mat, order){

  # Number of elements in the vector 'order'
  n <- length(order)

  # Check whether 'mat' is a 'matrix'
  if(!inherits(mat, "matrix")){
    stop("'mat' must be a matrix")
    # Check whether 'order' is of class 'character'
  } else if (!inherits(order, "character")){
    stop("'order' must be a character vector")
    # Check whether 'mat' is a symmetric matrix
  } else if(!(isSymmetric(mat))){
    stop("The matrix 'mat' must be symmetric")
    # Check whether 'order' has as many elements as there are rows
    # and columns in 'mat'
  } else if (n != length(colnames(mat))){
    stop("'order' must have as many elements as there are rows and
         columns in 'mat'")
    # Check whether the column names are in the 'order' vector
  } else if(length(which(colnames(mat) %in% order)) != n){
    stop("The column names of the matrix you want to reorder must
         be present in the vector 'order'")
    # Check whether the row names are in the 'order' vector
  } else if (length(which(row.names(mat) %in% order)) != n){
    print("The row names of the matrix you want to reorder must
          be present in the vector 'order'")
  } else {

    # Reorder 'mat' according to 'order'
    mat2 <- mat[order, order]

    return(mat2)
  }
}
```

Generating the grups (clusters) and reordering the symmetric matrix (dis. matrix):

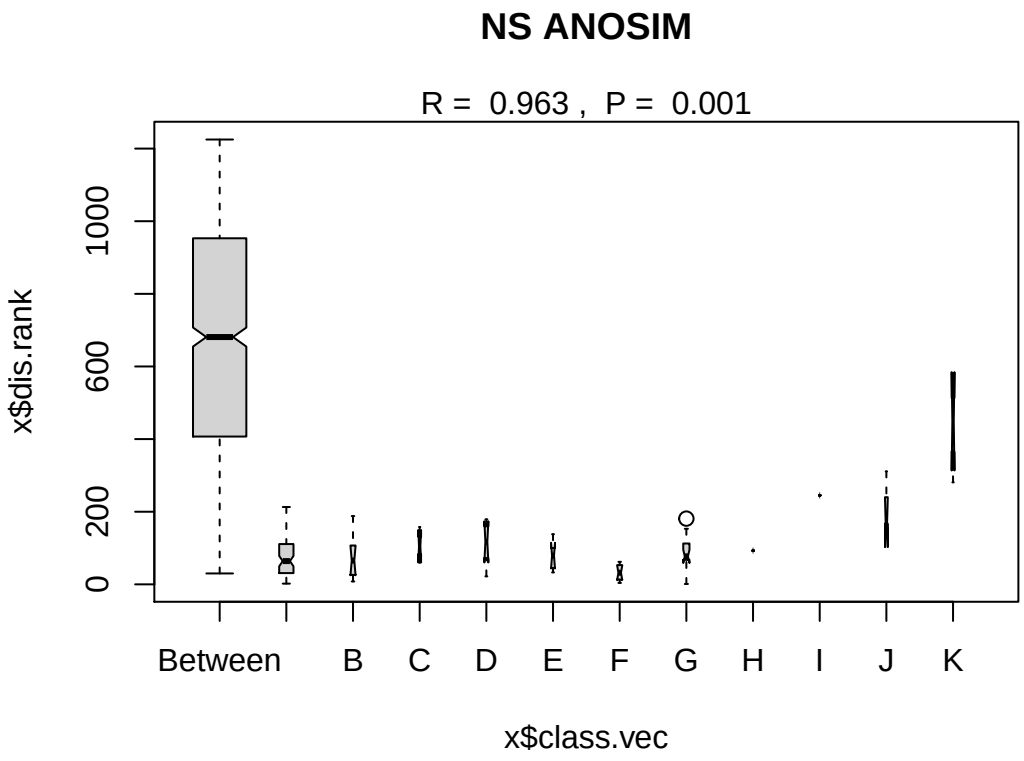
```
GO_ns_samples <- c("D1_S023_100L.m_R01", "D1_S320_716L.m_R00", "D1_S02_1L.s_R01", "D1_S02_10L.s_R03",
                   "D1_S023_10L.m_R01", "D1_S023_100L.m_R02", "D1_S02_10L.s_R01", "D1_S02_2.5L.s_R01.4"
                   "D1_S02_1L.s_R02", "D1_S02_2.5L.s_R01.2", "D1_S023_100L.m_R03", "D1_S023_60L.m_R01",
                   "D1_S023_60L.m_R03", "D1_S023_716L.m_R00", "D1_S023_496L.m_R00", "D1_S02_10L.m_R03",
                   "D1_S023_100L.m_R02.1", "D2_S023_100L.m_R11", "D1_S20_100L.m_R03", "D1_S20_496L.m_R0
                   "D1_S20_776L.m_R00", "D1_S20_120L.m_R01", "D1_S320_100L.m_R02", "D1_S02_10L.m_R01",
                   "D1_S320_100L.m_R02.1", "D2_S023_1000L.m_R00", "D1_S320_100L.m_R03", "D1_S02_1L.s_R0
                   "D1_S023_10L.m_R03", "D1_S02_10L.s_R02", "D1_S320_60L.m_R01", "D1_S02_2.5L.s_R01.3",
                   "D1_S02_2.5L.s_R01.1", "D1_S023_10L.m_R02", "D1_S320_60L.m_R03", "D2_S320_1000L.m_R0
                   "D1_S02_10L.m_R02", "D2_S320_100L.m_R11", "D1_S320_496L.m_R00", "D1_S320_100L.m_R01"
                   "D1_S20_100L.m_R02", "D1_S20_100L.m_R01", "D1_S320_10L.m_R01", "D1_S320_10L.m_R02",
                   "D1_S20_60L.m_R03", "D2_S20_100L.m_R11", "D2_S20_1000L.m_R00", "D1_S320_10L.m_R03",
                   "D1_S20_30L.m_R123", "D1_S20_100L.m_R02.1")
```

```
GO_ns_groups <- c(rep("A", 13), rep("B", 5), rep("C", 3), rep("D", 4), rep("E", 4), rep("F", 5), rep("G

GO_ns_GroupedSamples <- data.frame(GO_ns_samples, GO_ns_groups, row.names = 1)

GO_bray_mtx_reordered <- reorder_mat(GO_bray_mtx, GO_ns_samples)
```

ANOSIM Test:

```
GO_ns_anosim <- anosim(GO_bray_mtx_reordered, GO_ns_GroupedSamples$GO_ns_groups, permutations = 999, dis
par(cex=1, mar=c(5, 5, 5, 5))
plot(GO_ns_anosim, main="NS ANOSIM")
```

```
## Warning in bxp(list(stats = structure(c(30, 407, 681, 953, 1225, 2, 31, : some
## notches went outside hinges ('box'): maybe set notch=FALSE
```




```
summary(GO_ns_anosim) #ANOSIM significant
```

```
##
## Call:
## anosim(x = GO_bray_mtx_reordered, grouping = GO_ns_GroupedSamples$GO_ns_groups,      permutations = 9
## Dissimilarity: user supplied square matrix
##
## ANOSIM statistic R: 0.9633
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.103 0.135 0.159 0.186
```

```
##
## Dissimilarity ranks between and within classes:
##          0%    25%   50%    75% 100%    N
## Between  30 407.00 681.0 953.00 1225 1089
## A         2  31.25  64.0 110.75  213   78
## B         8  28.00  65.5 105.75  188   10
## C        60  82.50 105.0 131.50  158    3
## D        22  75.00 117.0 157.50  179    6
## E        33  51.00  79.0  97.25  138    6
## F         4  15.50  32.0  48.75   62   10
## G         1  67.50  77.0 112.50  181   15
## H        93  93.00  93.0  93.00   93    1
## I       245 245.00 245.0 245.00  245    1
## J       165 167.00 169.0 240.00  311    3
## K       281 365.00 449.0 513.50  578    3
```

ANOSIM significant

ANOSIM SIMKA

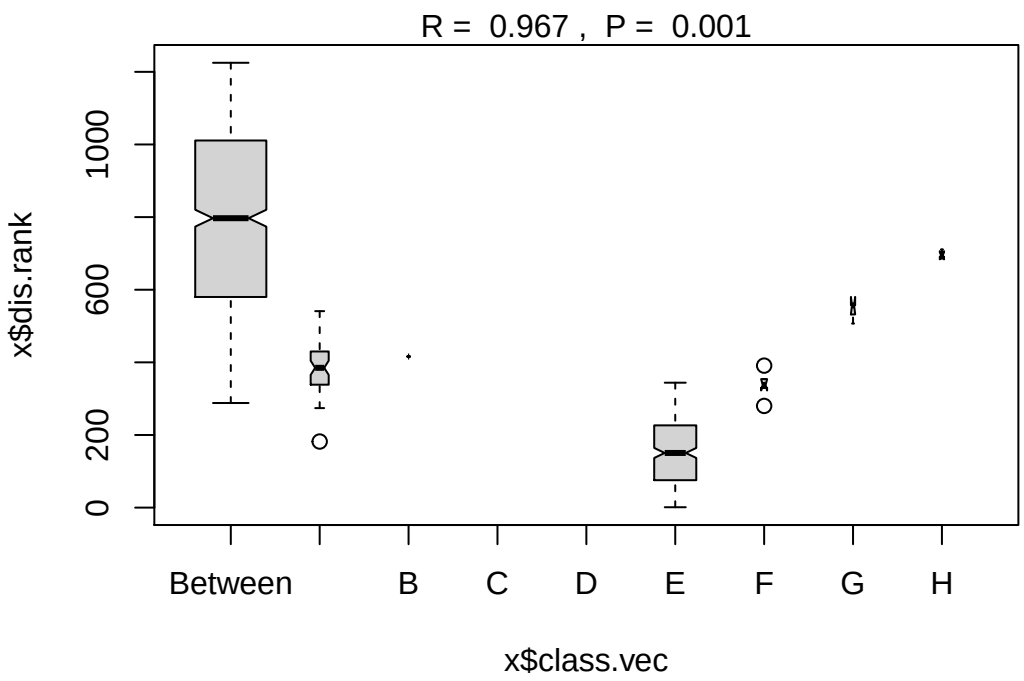
Generating the grups (clusters) and reordering the symmetric matrix (dis. matrix):

```
Simka_samples <- c("D1_S20_100L.m_R02.1", "D2_S20_1000L.m_R00", "D2_S20_100L.m_R11", "D1_S20_30L.m_R123
                   "D1_S20_776L.m_R00", "D1_S20_60L.m_R03", "D1_S20_100L.m_R02", "D1_S20_496L.m_R00",
                   "D1_S20_100L.m_R01", "D1_S20_120L.m_R01", "D1_S20_100L.m_R03", "D2_S320_1000L.m_R00"
                   "D2_S320_100L.m_R11", "D1_S320_496L.m_R00", "D1_S02_10L.m_R01", "D1_S02_10L.m_R02",
                   "D1_S023_10L.m_R03", "D1_S02_2.5L.s_R01.1", "D1_S02_2.5L.s_R01.3", "D1_S02_10L.s_R02
                   "D1_S023_10L.m_R02", "D1_S02_1L.s_R03", "D1_S02_10L.s_R03", "D1_S02_1L.s_R01",
                   "D1_S02_2.5L.s_R01.4", "D1_S02_10L.s_R01", "D1_S02_1L.s_R02", "D1_S02_2.5L.s_R01.2",
                   "D2_S023_1000L.m_R00", "D2_S023_100L.m_R11", "D1_S02_10L.m_R03", "D1_S023_100L.m_R01
                   "D1_S023_716L.m_R00", "D1_S023_60L.m_R01", "D1_S023_10L.m_R01", "D1_S023_496L.m_R00"
                   "D1_S023_100L.m_R02.1", "D1_S023_100L.m_R03", "D1_S023_100L.m_R02", "D1_S023_60L.m_R0
                   "D1_S320_716L.m_R00", "D1_S320_60L.m_R03", "D1_S320_60L.m_R01", "D1_S320_100L.m_R03"
                   "D1_S320_100L.m_R02", "D1_S320_100L.m_R01", "D1_S320_100L.m_R02.1", "D1_S320_10L.m_R0
                   "D1_S320_10L.m_R01", "D1_S320_10L.m_R02")

simka_mtx_reordered <- reorder_mat(simka_mtx, Simka_samples)

Simka_groups <- c(rep("A", 11), rep("B", 2), rep("C", 1), rep("D", 1), rep("E", 25), rep("F", 4), rep("G

Simka_GroupedSamples <- data.frame(Simka_samples, Simka_groups, row.names = 1)
```

ANOSIM Test:

```
Simka_anosim <- anosim(simka_mtx_reordered, Simka_GroupedSamples$Simka_groups, permutations = 999, dista
par(cex=1, mar=c(5, 5, 5, 5))
plot(Simka_anosim, main="SIMKA ANOSIM")
```

```
## Warning in bxp(list(stats = structure(c(288, 580, 797, 1011, 1225, 274, : some
## notches went outside hinges ('box'): maybe set notch=FALSE
```




```
summary(Simka_anosim) #ANOSIM significant
```

```
##
## Call:
## anosim(x = simka_mtx_reordered, grouping = Simka_GroupedSamples$Simka_groups,      permutations = 999
## Dissimilarity: user supplied square matrix
##
## ANOSIM statistic R: 0.9669
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##   90%   95% 97.5%   99%
## 0.105 0.143 0.175 0.218
##
## Dissimilarity ranks between and within classes:
##          0%    25%   50%     75% 100%   N
## Between 288 580.00 797.0 1011.00 1225 857
## A       182 338.50 385.0  430.00  541  55
## B       416 416.00 416.0  416.00  416   1
## C        NA     NA    NA      NA   NA   0
## D        NA     NA    NA      NA   NA   0
## E         1  75.75 150.5  226.25  344 300
## F       280 331.50 338.0  350.50  391   6
## G       507 532.00 557.0  557.50  558   3
## H       686 690.50 695.0  703.00  711   3
```

ANOSIM significant