01_metho_comparision_taxo

Mathis Gheno

2024-09-20

---

load required packages

```
library(tidyverse)
library(ade4) # Coinertia
library(vegan) # Rarefaction
library(ggrepel)
library(metacoder)
```

```
## Warning: le package 'metacoder' a été compilé avec la version R 4.3.3
```

```
main_theme = theme_minimal()+
  theme(line = element_blank(),
        axis.line = element_line(colour = "black"),
        panel.border = element_blank(),
        axis.ticks =  element_line(colour = "black"),
        axis.text.x = element_text(colour = "black", size=22, face="italic", angle = 45, vjust = 1, hju
        axis.text.y = element_text(colour = "black", size=22, face="italic"),
        legend.title = element_text(colour = "black", size=20,
                                    hjust =0.5),

        legend.text = element_text(colour = "black", size=18),
        axis.title= element_text(size=28),
        strip.text = element_text(colour = "black", size=15, face ="italic"))
```

Data path

```
metab1 = "data/metabarcoding_vs_metagenomics/oracle_metaBt0_16S_samples_run_20190715_kraken2_assignment
#metab2 = "data/metabarcoding_vs_metagenomics/oracle_metaBt0_16S_samples_run_20200106_kraken2_assignmen
metag1 = "data/metabarcoding_vs_metagenomics/oracle_metaGt0_SAMA_12_samples_kraken2_SSU_assignment_genu
#metag2 = "data/metabarcoding_vs_metagenomics/oracle_metaGt0_SAMA_21_1_samples_kraken2_SSU_assignment_g
#metag3 = "data/metabarcoding_vs_metagenomics/oracle_metaGt0_SAMA_21_2_samples_kraken2_SSU_assignment_g
```

Load data

Set up function to load and pepare data

```
taxonomic_levels <- c("kingdom", "phylum", "class", "order", "family",
                      "genus", "species")
## ---- Arrange metaB ----
```

```
# Select dat from bracken method, and rename columns
select_and_rename_cols = function(tab, method){
  tab%>%
    read_tsv(col_names = T, skip = 1)%>%
    select(which(str_detect(colnames(.),"bracken_genuses")), "#OTU ID", "taxonomy")%>%
    rename_with( # rename colnames
      .%>%
      str_remove( "_bracken_genuses")%>%
      str_replace_all("^OR-", "BU_") %>%
      str_replace_all("-", "_")%>%
      str_remove("_S\\d+")%>%
      str_replace("((?<!.)T_)(\\d)", "CONT_BU_PCR_\\2")%>%
      str_replace("(BU_T_extr)", "CONT_BU_ext")%>%
      str_replace("#OTU ID", "OTU")
      )%>%rename_with(~ paste0(., "_",method), contains("BU"))%>%
    separate_wider_delim(taxonomy, names = taxonomic_levels, delim = "; ", cols_remove = F)%>% # Separa
    filter(kingdom == "k__Bacteria")
  }
```

Decontamination function

Compute the total number of reads for each OTU present in control samples. That sum is then substracted from all occurrences of that OTU in true samples. The rationale is as follows:

- if an OTU is abundant in control samples, but rare in true samples, then it is a contamination specific to the control samples, and it will be eliminated by the substraction (i.e, final abundance is 0),
- if an OTU is present in control samples, and present in true samples (systematic contamination, will be mitigated by the substraction),
- if an OTU is rare in control samples, but abundant in true samples (cross-talk, will be eliminated/mitigated by the substraction)

Control samples can be eliminated from the statistical analysis after the substraction (all OTUs present in control samples have been zeroed out).

```
.%>%
  #select(!ends_with(run_rm))%>%
  replace(. == 0, NA) %>%
  pivot_longer(starts_with("BU"), names_to = "samples", values_to = "reads") %>%
  filter(!is.na(reads)) %>%
  #{{merge control samples}}
  mutate( n = rowSums(across(starts_with("CONT")), na.rm = T))%>%
  #{{substract abundance of control samples}}
  mutate(reads = case_when(
    is.na(n)  ~ reads,
    n > reads ~ 0,
    TRUE      ~ reads - n)) %>%
  select(-n,-starts_with("CONT"))%>%
  pivot_wider(values_from = reads, names_from = samples, values_fill = 0) -> decontaminate
```

Load and format data for coinertia

```
metab1%>%
 select_and_rename_cols("B")%>%
 decontaminate -> metab1_decont_table
```

```
## Rows: 3726 Columns: 332
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr   (1): taxonomy
## dbl (331): #OTU ID, OR-RT-10_S40_R1, OR-RT-10_S40_R1_bracken_genuses, OR-RT-...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
metag1%>%
 select_and_rename_cols("G")%>%
 decontaminate -> metag1_decont_table
```

```
## Rows: 4826 Columns: 318
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr   (1): taxonomy
## dbl (317): #OTU ID, BU-RT-01_R1, BU-RT-01_R1_bracken_genuses, BU-RT-01_R2, B...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
metag1_decont_table%>%
  full_join( metab1_decont_table, by = join_by(OTU == OTU, kingdom==kingdom, phylum == phylum,
                                              class == class, order == order, family == family,
                                              genus == genus,  species == species, taxonomy == taxonomy
   mutate(across(where(is.numeric), ~ replace(., is.na(.), 0)))-> meta_g_b_decond_table
```

```
metab1_decont_table%>%
  select(starts_with("BU"))%>%rowSums
```

```
##   [1]  63734  69097  31150  74201   1659   1302   4699    156   4307    593
##  [11]   1283    142   1274    247   1777     62    138  94049   2426    526
##  [21]   1883  11421    855   2712   1247  62606  15173   3512    664     87
##  [31]  23874  52292   6370  67849  32693 138435   2570   4504    734  10511
##  [41]   1608 221134    841  57687  47796  15607  14548  14134   2627   2647
##  [51]    104   8760   3660  11594   2614   1231    920   1242   8372     14
##  [61]    979  99454   9936   6529   1017   9052  15793    768     34   2269
##  [71]   3370    708   1507   1024   2440    196    443     85   1142    163
##  [81]   1263    797    304   1727    238    542    102    560     50   4057
##  [91]   2782     34     13    966     36  37035  13312   2011   1994   8481
## [101]   3723  11419   2713   3494    828   2554  35550    204    155    311
## [111]  49310    461     55   1447   2061     11    261 217582  84571   7891
## [121]   4217   1103   2003 475437    170 793741  12703   4128     23   3834
## [131]   1613     80   7003  11257   1324    358     95 180269  39281  16325
## [141]  12557    272   1665    467     38     60 109974  23609  27000  24495
## [151]    472   1603   5036    388  10675   1341    508   4902      0   2223
## [161]    418     17   7880  10087   2615   3122   4634  20805     51     16
## [171] 106048   2013      1   2706    357   1848    160   1784  21484   7638
## [181]  17454   2471     79    361   1394    447    346    515    158  20449
## [191]   6248   4674   3849   7188   1442     37  39851   3677     62    335
## [201]    479   4479    304   5729   5909   5169    951  72974  11665    707
## [211]    250    196    199     19   2631   1079     16     65     30   4145
## [221]  87752  38946  10558  28134  33574  50753   6112      0     11     17
## [231]  25842   2665   5052   7255   4279  32124   3660   2262    156   4981
## [241]    356     24   4713    303    736     77     55    391     27   4218
## [251]    536    336     54      0    868    178    164    262    226   1221
## [261]    796     46  25932   3404  19177      0      0   3104    778     94
## [271]     95    170   3286    871     84   1527    526   1531     53    622
## [281]     94    161 315454   2242   1240   9070   3454   1805   1716    304
## [291]   1796   7514   1369    219  87219  44541  84591   7429   9634   3197
## [301]   4981  10549    345  21325   9646 183714    203   5922    530   4279
## [311]   3589   3979   1704 565448  10956  15091  15169   3609   7372     82
## [321]    562    328    183   6942  22512  27136  12842   7812   1312   6033
## [331]    116   9081    192    185   1010   7034  80163 125392   9737   7759
## [341]    127   1550   1099  50042    639  35960   3773   5302   7893     97
## [351]   7659   3359    678     89   3703     34    140  10964     83    396
## [361]   1221   2210     15    861   4980   8934    934    164    424     30
## [371]   1531     90    856    984    202    373     88   2139  38236    736
## [381]    497     89     75     44   1990     44    865    107     38    546
## [391]   6294  62338  18941  25512  45194     34   1368     18   5048 119190
## [401]  44711   3637 290077   4886    319   4314   1319   1631  14612 186185
## [411]  28497   1832  19735   1416   1962    342     16     41  12697   9310
## [421]   4842    152   1151    516   1082     33    976 264589  42443  51361
## [431]  16116    354  10196   3163   2132   1922    690     87     47   8046
## [441]    393   6177  62751    117  67285   6563  21067      3    974    665
## [451]   8052    218    740  23134   1669  17801  16924   6193   1712   2213
## [461]   2037      0   3438    840    775    338    107     12    541    293
## [471]    107    637     67   6473    813    101   6491     11    218   1128
## [481]    201    134    535     14     33   5753  26842   4727  26296   3527
## [491]    251   1134    479     38     23 165288   9322   2568   3748    124
## [501]     57     30   1927     83    432    822  17167    916   2827    825
## [511]    810    423    231     81    289     42     71   4089   4276   1390
## [521]    360     65   1295     66     47     85     15   5827     11   6339
## [531]     88    579     14    786    175     27    673    347    132     49
## [541]     41    349    168    234     82    793     94   5851   1724    237
## [551]  55276   3098   2694  18533     19    163    145   1142   1959   1682
## [561]    203    530     10    588    167    239    922    339     39    381
## [571]    195    305    374     52    146     79     79    727    205    234
## [581]   1920     61    477    114    237    124     46   1746     23     39
## [591]    156     19    165     15    196    888     58     96     80     13
## [601]    606    392     46     13      0      0     71     22   1394    274
## [611]     75     28    684   1071    513    344   1009    789     80    301
## [621]    140   2329     16     60    250     31    133     59    187     52
## [631]    533     24    942     61    303     94     19     52     25     12
## [641]    622     42     86     20     30     13     59    306     56     65
## [651]    941    324     33     23    174     26    706     60     26    528
## [661]    360    569    267    139     61     29     15     41    565     86
## [671]     31     75     19     91      0     91    174     96   1547    487
## [681]   1603   2157    171     86     95    237     38     36     48    341
## [691]    344     10     57     15     14     19    370     12     13     29
## [701]     21    107    260     33    254     15    211     72     24     21
## [711]     27    657     50    145     21     10    258     64    194    477
## [721]     82    144    200     60    579    452     11     84    106     33
## [731]    163     17    274     11     67    768    257     64    216    205
## [741]     12    233    230      0     36    197     12     66     66    439
## [751]     57     93     43     29     10     77     32    189     41    104
## [761]    160    158   1473     30     10     12    356    191     64     34
## [771]    135     22     22     95     61     47     73     28     15    196
## [781]      0     39    102     35    102     39    443     32     13    103
## [791]     18     41     25    454     22     14    105    234    135    331
## [801]     39     87     13    477    268    143     23    382     13     37
## [811]    110    158     36     10     19     20     14    269     21     16
## [821]     33     19     37     12
```

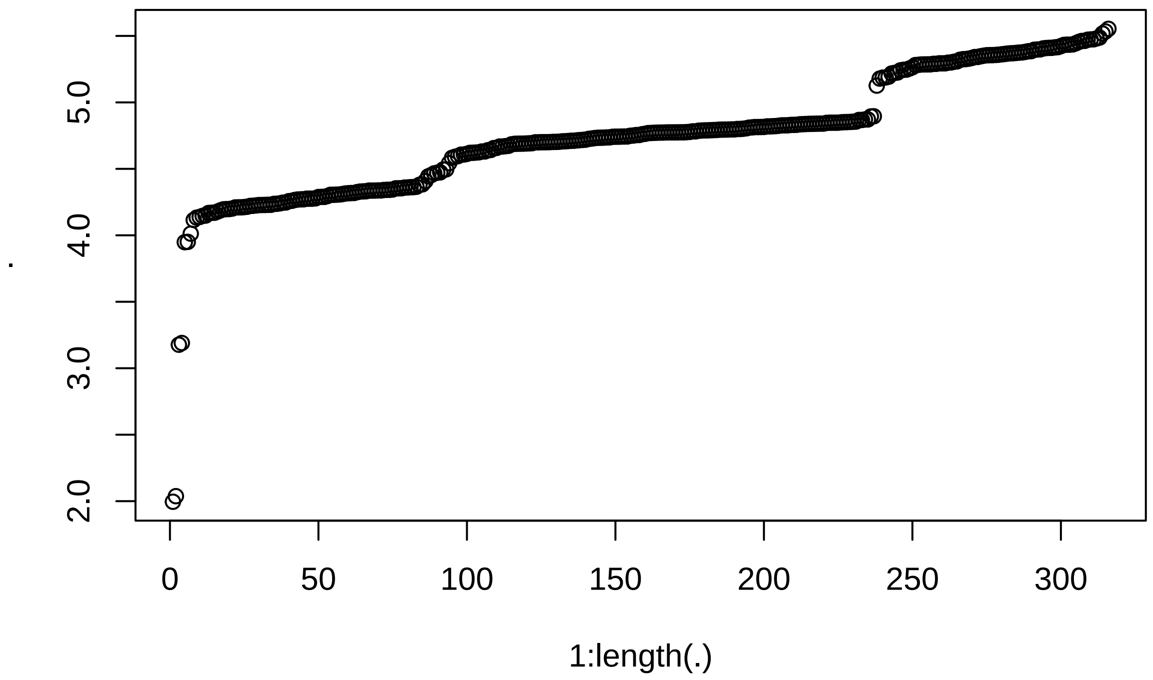
Is it normal that some OTU have 0 reads ?

Rarefaction function

```
rarefaction.func = function(df,rar_sample, rarcur = T){


  if(rarcur){ # {{ if rarefaction curve needed}}
    metab1_decont_table%>%
  select(starts_with("BU"))%>%
  t()%>%rarecurve(step = 20, sample = rar_sample, col = "blue", cex = 0.6)
  }
   # {{rarefaction}}
  df%>%
    select(starts_with("BU"))%>%
    t()%>%
    rrarefy(rar_sample)%>%
    t()%>%
    bind_cols(
      df%>%select(!starts_with("BU"))
    )-> df_rarefy
  return(df_rarefy)
}
```

Which value to rarefy ?

```
meta_g_b_decond_table%>%
  select(starts_with("BU"))%>%
  t()%>%rowSums%>%sort%>%log10%>%plot(1:length(.),.)
```

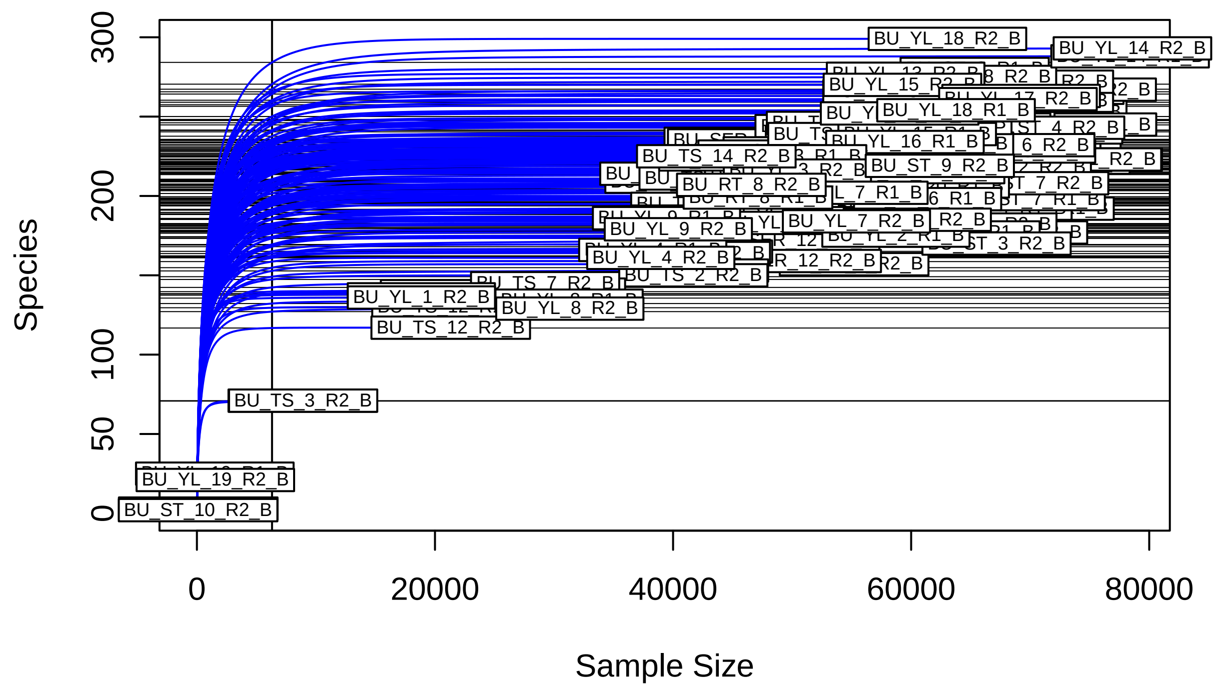


Gap around 10^3.8 => rarefy at this value

Rarefaction

```
meta_g_b_decond_table%>%rarefaction.func(rar_sample =round(10^3.8), rarcur = T) -> meta_g_b_decond_tabl
```

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



##Coinertia

Fist PCA for each method

```
meta_g_b_decond_table_rare%>%
  column_to_rownames("OTU")%>%
  select(ends_with("R1_B"))%>%
  select(order(colnames(.)))%>%
  decostand(method = "hellinger")%>%
  dudi.pca( scale = TRUE, scan = FALSE, nf = 3) -> dudi_b_r1

meta_g_b_decond_table_rare%>%
  column_to_rownames("OTU")%>%
  select(ends_with("R2_B"))%>%
  select(order(colnames(.)))%>%
  decostand(method = "hellinger")%>%
  dudi.pca( scale = TRUE, scan = FALSE, nf = 3) -> dudi_b_r2

meta_g_b_decond_table_rare%>%
  column_to_rownames("OTU")%>%
  select(ends_with("R1_G"))%>%
  select(order(colnames(.)))%>%
  decostand(method = "hellinger")%>%
  dudi.pca( scale = TRUE, scan = FALSE, nf = 3) -> dudi_g_r1

meta_g_b_decond_table_rare%>%
  column_to_rownames("OTU")%>%
  select(ends_with("R2_G"))%>%
  select(order(colnames(.)))%>%
  decostand(method = "hellinger")%>%
  dudi.pca( scale = TRUE, scan = FALSE, nf = 3) -> dudi_g_r2
# Disjoint R1 and R2 and decontaminate
```

I'm doing the Hellinger transformation on the matrix (OTU x site). Usually the transformation (and the coinertia by extension) is done on the transpose of this matrix (site x species/OUT).It is because most of the time we are interested in the difference of species composition in sites. However here we are interested in the species detection difference, in other word we are looking for difference of species abundances in sites

/! Maybe do separate or merge Hellinger transformation on the 2 df ?

Better graphical representation function

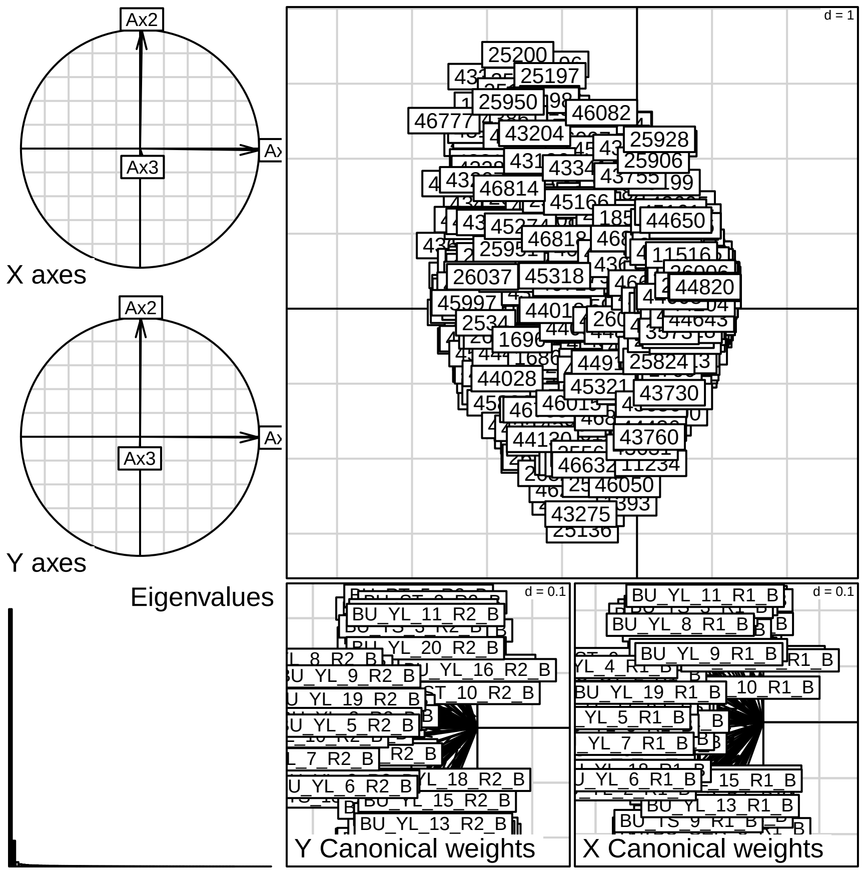
```
### ---- better graph representation for coinertia
coin.graph = function(coin_obj,meta_obj, brin, method){
  # {{recover coordinate in the ordination space of the 2 method to compaire}}
  cbind(coin_obj$mX,coin_obj$mY) ->  df_coin_pos
  colnames(df_coin_pos)=c("metho1_x", "metho1_y", "metho2_x", "metho2_y")

  df_coin_pos%>%
    rownames_to_column("OTU")%>%
    mutate(OTU = as.numeric(OTU))%>%
     # {{recover OTU and genus information}}
    left_join(meta_obj%>%select(OTU, genus), by = "OTU")%>%
    # {{Find taxa with the higher and lower diff (lower and higer distance in the ordination space betw
    mutate(dist =  sqrt(rowSums((coin1$mX-coin1$mY)^2)))%>%
    arrange(desc(dist)) %>%
    slice(c(1:10, (n() - 9):n()))%>%
    # {{add a colums to facet_wrap}}
    mutate(diff = c(rep("Strong",10 ), rep("Weak", 10)))%>%

    ggplot()+
    facet_wrap(~diff)+
    geom_label_repel(aes(x= metho1_x, y = metho1_y ,label = genus))+
    geom_segment(aes(x = metho1_x, y = metho1_y, xend = metho2_x, yend = metho2_y),
                 arrow = arrow(length = unit(0.3, "cm"), type = "closed"), cex =1)+

    labs(x="X",y = "Y", title = paste("Genera estimations differences between", brin[1],method[1], "(ar
    main_theme
    }
```

Metabarcoding R1 vs R2

```
coin1 <- coinertia(dudi_b_r1,dudi_b_r2, scan = FALSE, nf = 2)

summary(coin1)
```

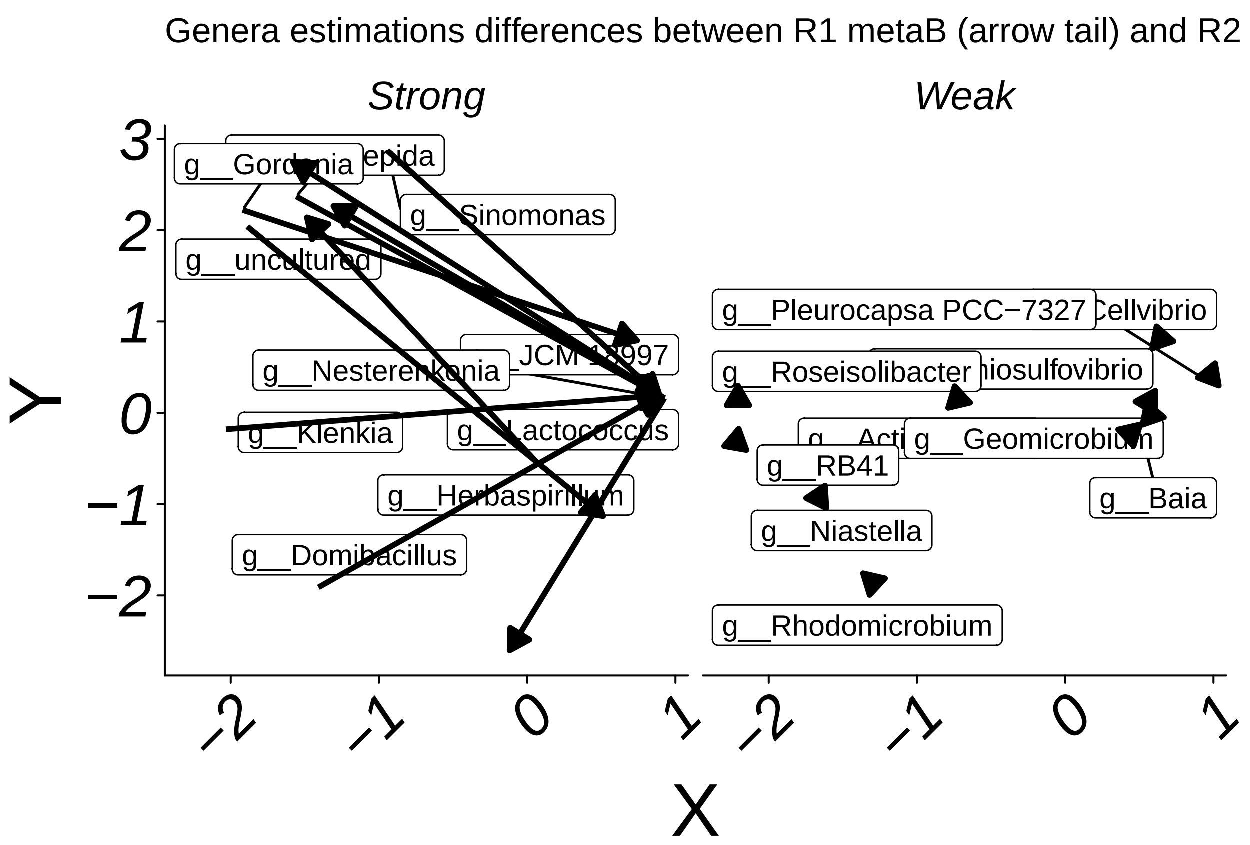
```
## Coinertia analysis
##
## Class: coinertia dudi
## Call: coinertia(dudiX = dudi_b_r1, dudiY = dudi_b_r2, scannf = FALSE,
##     nf = 2)
##
## Total inertia: 184
##
## Eigenvalues:
##     Ax1     Ax2     Ax3     Ax4     Ax5
## 152.365  15.260   2.574   1.337   1.079
##
## Projected inertia (%):
##     Ax1     Ax2     Ax3     Ax4     Ax5
## 82.7853  8.2914  1.3984  0.7265  0.5863
##
## Cumulative projected inertia (%):
##     Ax1   Ax1:2   Ax1:3   Ax1:4   Ax1:5
##   82.79   91.08   92.48   93.20   93.79
##
## (Only 5 dimensions (out of 79) are shown)
##
## Eigenvalues decomposition:
##         eig     covar      sdX      sdY      corr
## 1 152.36540 12.343638 3.807983 3.992351 0.8119316
## 2  15.26015  3.906424 2.219802 2.220008 0.7927031
##
## Inertia & coinertia X (dudi_b_r1):
##     inertia      max     ratio
## 1  14.50073 14.55301 0.9964077
## 12 19.42825 19.55067 0.9937383
##
## Inertia & coinertia Y (dudi_b_r2):
##     inertia      max     ratio
## 1  15.93887 15.99369 0.9965723
## 12 20.86731 20.98130 0.9945671
##
## RV:
##  0.587031
```

```
plot(coin1)
```



```
coin.graph(coin1,meta_obj= meta_g_b_decond_table_rare, brin = c("R1","R2"), method = c("metaB", "metaB"
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



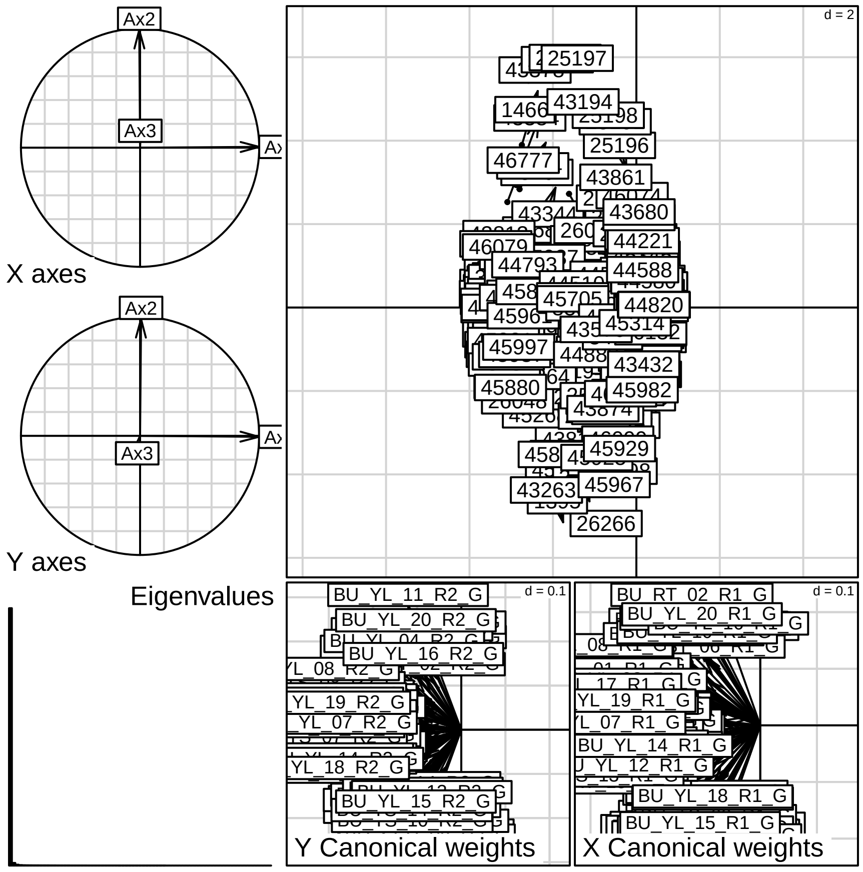
Metagenomic R1 vs R2

```
coin2 <- coinertia(dudi_g_r1,dudi_g_r2, scan = FALSE, nf = 2)

summary(coin2)
```

```
## Coinertia analysis
##
## Class: coinertia dudi
## Call: coinertia(dudiX = dudi_g_r1, dudiY = dudi_g_r2, scannf = FALSE,
##     nf = 2)
##
## Total inertia: 725.2
##
## Eigenvalues:
##      Ax1       Ax2       Ax3       Ax4       Ax5
## 704.4592   7.1899   2.3179   1.1892   0.8319
##
## Projected inertia (%):
##     Ax1     Ax2     Ax3     Ax4     Ax5
## 97.1464  0.9915  0.3196  0.1640  0.1147
##
## Cumulative projected inertia (%):
##     Ax1   Ax1:2   Ax1:3   Ax1:4   Ax1:5
##   97.15   98.14   98.46   98.62   98.74
##
```

```
## (Only 5 dimensions (out of 77) are shown)
##
## Eigenvalues decomposition:
##          eig     covar      sdX      sdY      corr
## 1 704.459226 26.541651 5.593758 5.441799 0.8719304
## 2   7.189869  2.681393 1.886951 1.615688 0.8795134
##
## Inertia & coinertia X (dudi_g_r1):
##     inertia      max     ratio
## 1  31.29012 31.31363 0.9992495
## 12 34.85071 34.93586 0.9975626
##
## Inertia & coinertia Y (dudi_g_r2):
##     inertia      max     ratio
## 1  29.61317 29.72186 0.9963431
## 12 32.22362 32.39304 0.9947697
##
## RV:
##  0.7402294
```

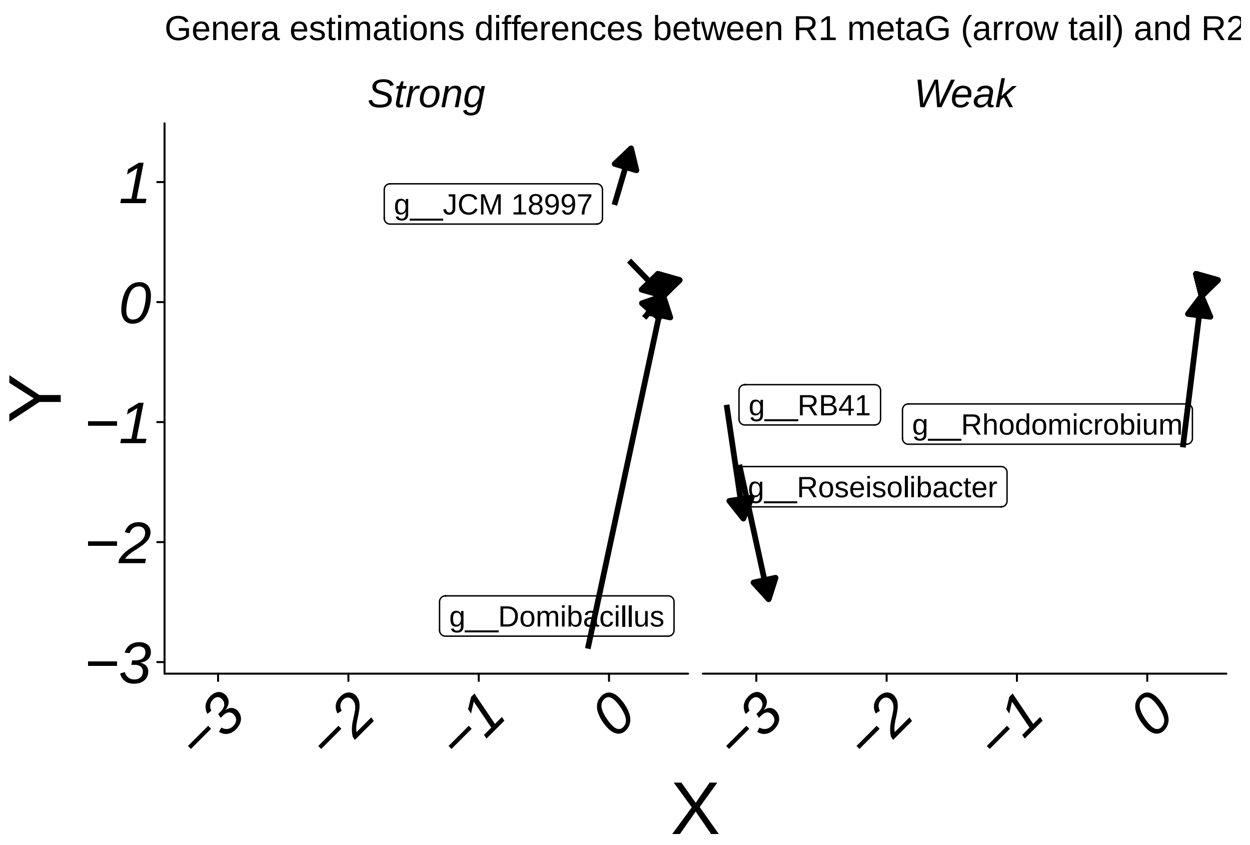
```r
plot(coin2)
```



```r
coin.graph(coin2,meta_obj= meta_g_b_decond_table_rare, brin = c("R1","R2"), method = c("metaG", "metaG"
```

```
## Warning: ggrepel: 8 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

```
## Warning: ggrepel: 7 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

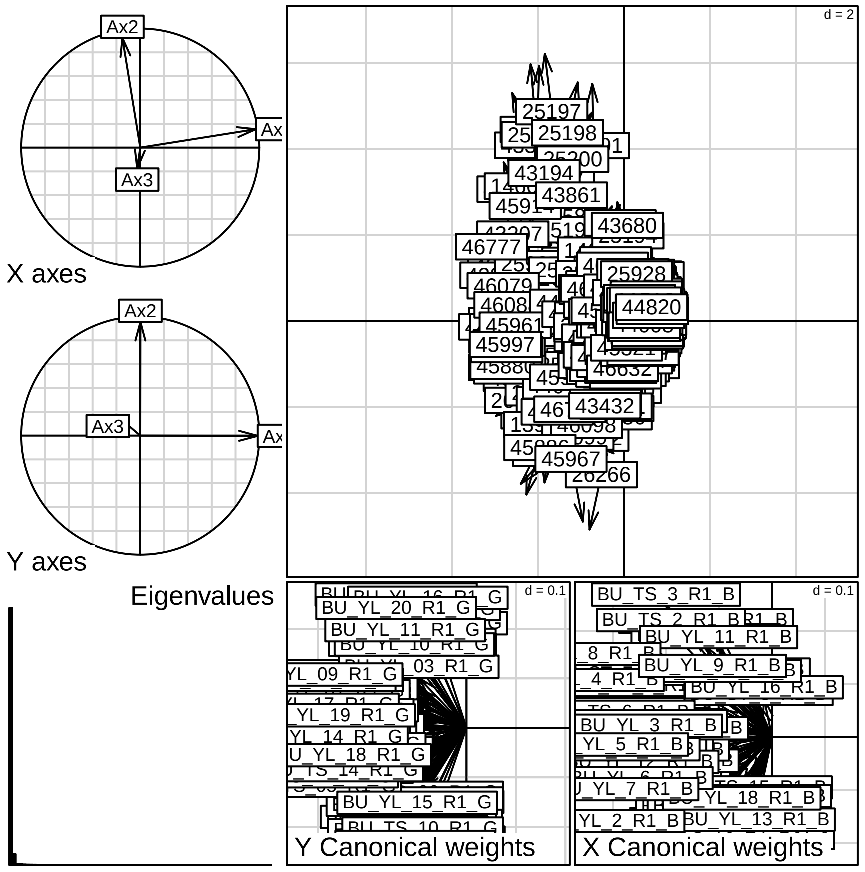


Metabarcoding R1 vs Metagenomic R1

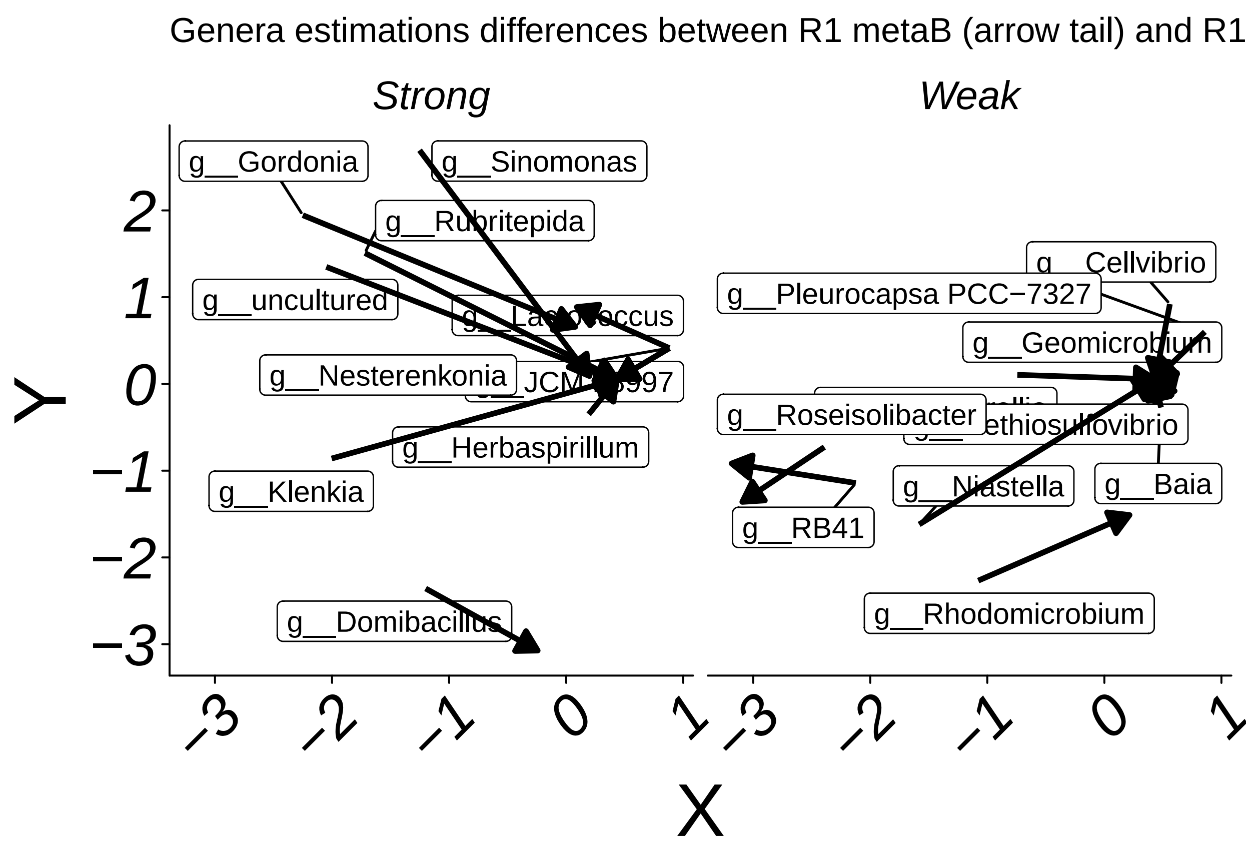
```r
coin3 <- coinertia(dudi_b_r1,dudi_g_r1, scan = FALSE, nf = 2)

summary(coin3)
```

```
## Coinertia analysis
##
## Class: coinertia dudi
## Call: coinertia(dudiX = dudi_b_r1, dudiY = dudi_g_r1, scannf = FALSE,
##     nf = 2)
##
## Total inertia: 90.71
##
## Eigenvalues:
##     Ax1     Ax2     Ax3     Ax4     Ax5
## 84.6078  3.5969  0.4025  0.3085  0.2136
##
## Projected inertia (%):
##     Ax1     Ax2     Ax3     Ax4     Ax5
## 93.2709  3.9652  0.4437  0.3401  0.2354
##
## Cumulative projected inertia (%):
##     Ax1   Ax1:2   Ax1:3   Ax1:4   Ax1:5
##   93.27   97.24   97.68   98.02   98.26
##
## (Only 5 dimensions (out of 76) are shown)
##
## Eigenvalues decomposition:
##         eig    covar      sdX      sdY      corr
## 1 84.607820 9.198251 3.725575 5.520419 0.4472391
## 2  3.596948 1.896562 2.187440 1.840701 0.4710289
##
## Inertia & coinertia X (dudi_b_r1):
##     inertia      max     ratio
## 1  13.87991 14.55301 0.9537484
## 12 18.66481 19.55067 0.9546888
##
## Inertia & coinertia Y (dudi_g_r1):
##     inertia      max     ratio
## 1  30.47503 31.31363 0.9732195
## 12 33.86321 34.93586 0.9692966
##
## RV:
##  0.1650627
```

```r
plot(coin3)
```



```r
coin.graph(coin3,meta_obj= meta_g_b_decond_table_rare, brin = c("R1","R1"), method = c("metaB", "metaG"
```



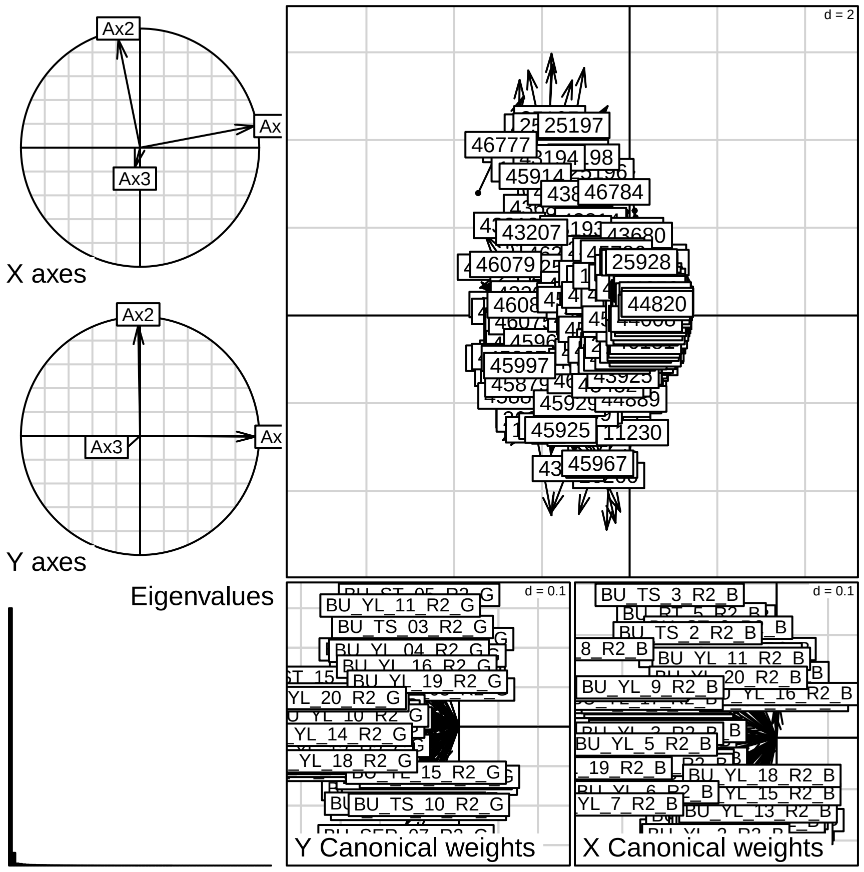
Metabarcoding R2 vs Metagenomic R2

```
coin4 <- coinertia(dudi_b_r2,dudi_g_r2, scan = FALSE, nf = 2)

summary(coin4)
```

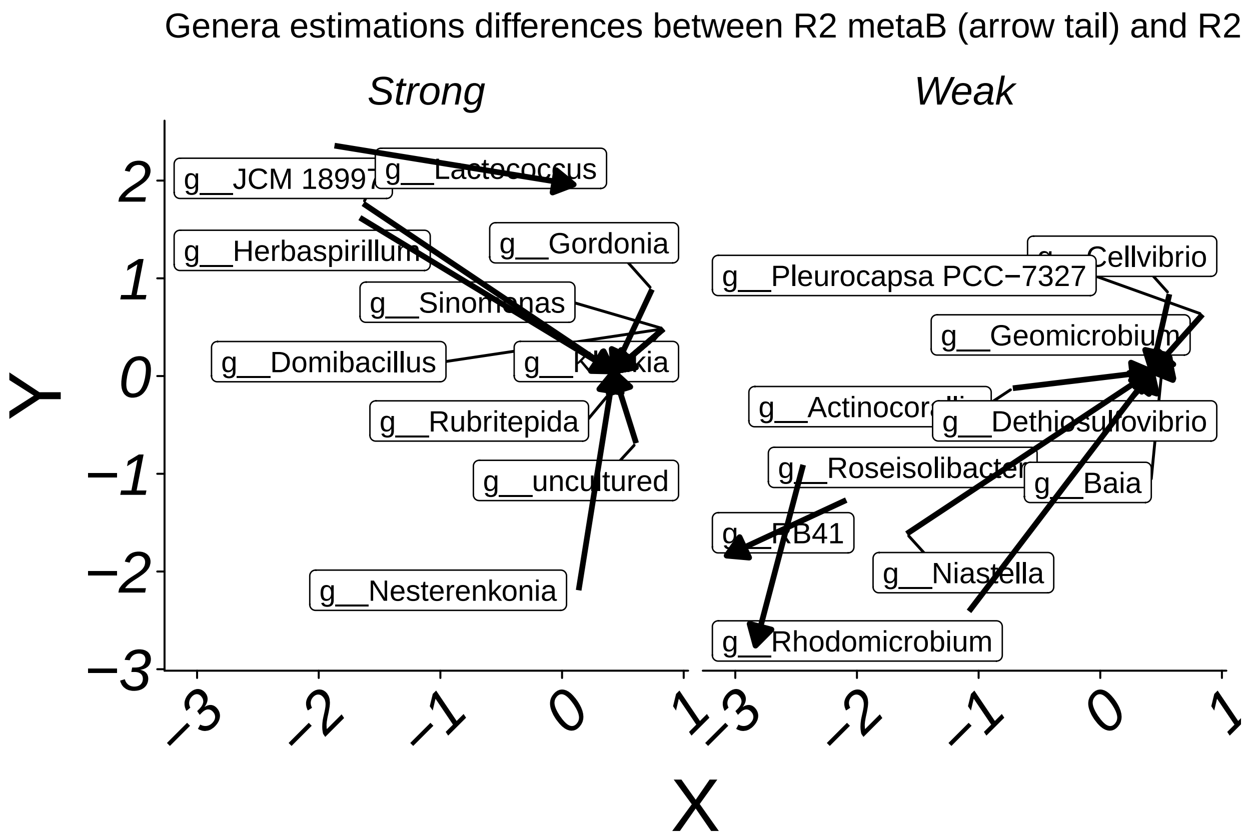
```
## Coinertia analysis
##
## Class: coinertia dudi
## Call: coinertia(dudiX = dudi_b_r2, dudiY = dudi_g_r2, scannf = FALSE,
##     nf = 2)
##
## Total inertia: 55.28
##
## Eigenvalues:
##     Ax1     Ax2     Ax3     Ax4     Ax5
## 50.2480  2.4825  0.3951  0.3185  0.2346
##
## Projected inertia (%):
##     Ax1     Ax2     Ax3     Ax4     Ax5
## 90.9029  4.4911  0.7147  0.5763  0.4244
##
## Cumulative projected inertia (%):
##     Ax1   Ax1:2   Ax1:3   Ax1:4   Ax1:5
##   90.90   95.39   96.11   96.68   97.11
##
```

```
## (Only 5 dimensions (out of 77) are shown)
##
## Eigenvalues decomposition:
##         eig    covar     sdX      sdY      corr
## 1 50.248016 7.088583 3.861538 5.278733 0.3477519
## 2  2.482527 1.575604 2.195488 1.557917 0.4606507
##
## Inertia & coinertia X (dudi_b_r2):
##     inertia      max     ratio
## 1  14.91148 15.99369 0.9323348
## 12 19.73164 20.98130 0.9404395
##
## Inertia & coinertia Y (dudi_g_r2):
##     inertia      max     ratio
## 1  27.86502 29.72186 0.9375260
## 12 30.29212 32.39304 0.9351429
##
## RV:
##  0.09890585
```

```r
plot(coin4)
```



```r
coin.graph(coin4,meta_obj= meta_g_b_decond_table_rare, brin = c("R2","R2"), method = c("metaB", "metaG"
```



Metacoder

Function that make the metacoder object that contain the taxonomic comparison tree

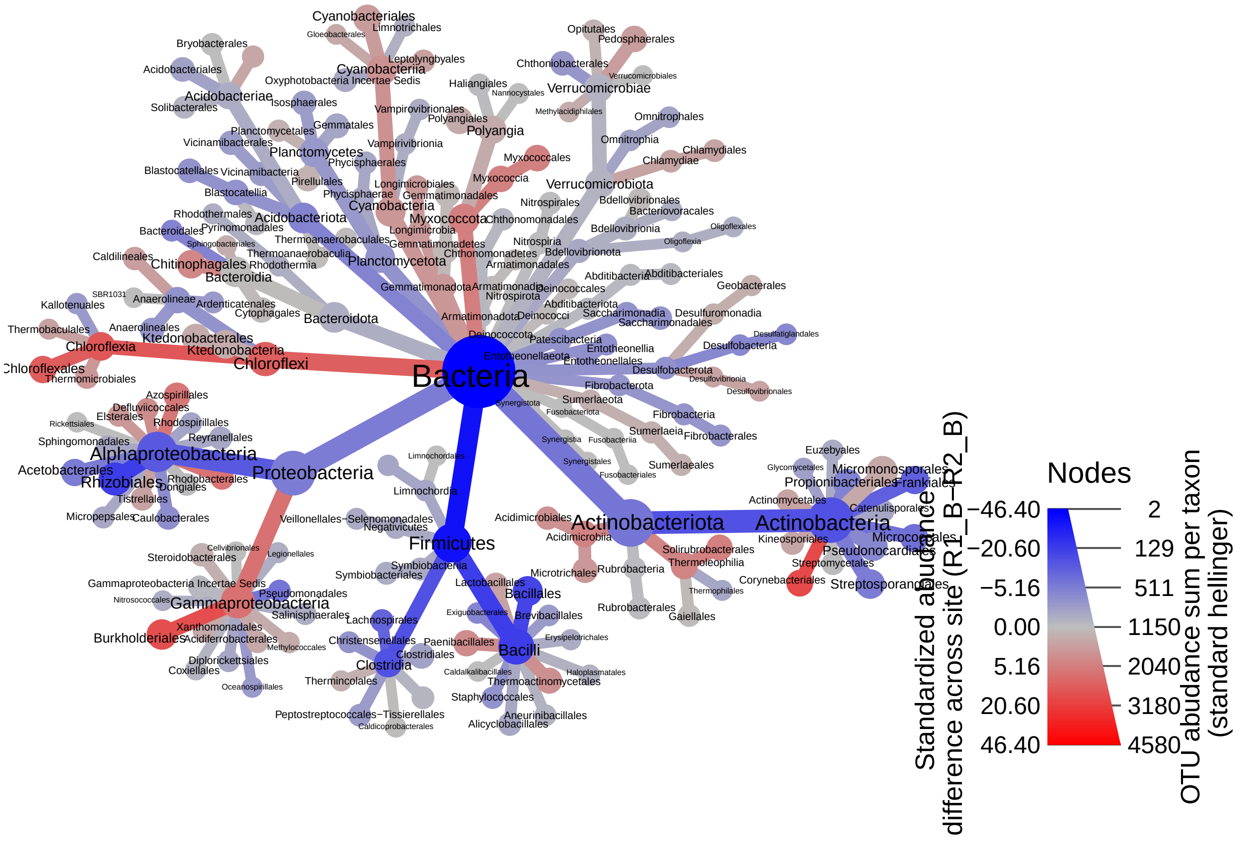
```
make.tax.tree.comp = function(df,method){
  df %>%
  select(OTU, kingdom, phylum, class, order, family, genus, species, taxonomy, ends_with(method[1]), en
  # {{standardization of the table produce by the "method 1" (first method put in the vector)}}
  mutate(across(ends_with(method[1]), ~ sqrt(. / rowSums(across(ends_with(method[1]))))), .names = "hell
  # {{standardization of the table produce by the "method 2" (second method put in the vector)}}
  across(ends_with(method[2]), ~ sqrt(. / rowSums(across(ends_with(method[2])))), .names = "hellinger_{
  # {{remove species that don't appear in the 2 method that we compare in this chunk}}
  mutate(total = rowSums(across(starts_with("hellinger_") )))%>%
  filter(total !=0 )%>%
  # {{create metacoder object}}
  metacoder::parse_tax_data(class_cols = "taxonomy", # The column in the input table
                    class_sep = "; ",
                    class_regex = "^([a-z]{0,1})_{0,2}(.*)$",
                    class_key = c("tax_rank" = "taxon_rank", "name" = "taxon_name")) -> obj

  # {{compute abudance (standadize by hellinger) per taxon for the 2 methode}}
  obj$data$tax_abund <- metacoder::calc_taxon_abund(obj, "tax_data",
                                          cols = startsWith(colnames(obj$data$tax_data),"hellinger"),
                                                            groups = str_extract(str_subset(colnames(obj$da
  return(obj)
}
```

```
meta_g_b_decond_table_rare%>%
  make.tax.tree.comp(method = c("R1_B","R2_B")) -> obj1
```

```
## Summing per-taxon counts from 158 columns in 2 groups for 1519 taxa
```

Metabarcoding R1 vs R2

```
obj1%>%
  filter_taxa(taxon_ranks == "o", supertaxa = TRUE)%>%
  heat_tree(node_label = gsub(pattern = "\\[|\\]", replacement = "", taxon_names),
            node_size = R1_B+R2_B,
            node_color = R1_B-R2_B,
            node_color_range = c("blue", "gray", "red"),
            node_color_interval = c(-max(abs(R1_B-R2_B)), max(abs(R1_B-R2_B))),
            node_color_axis_label = "Standardized abudance \n difference across site (R1_B-R2_B)",
            node_size_axis_label = "OTU abudance sum per taxon\n (standard hellinger)",
            layout = "davidson-harel", initial_layout = "reingold-tilford")
```



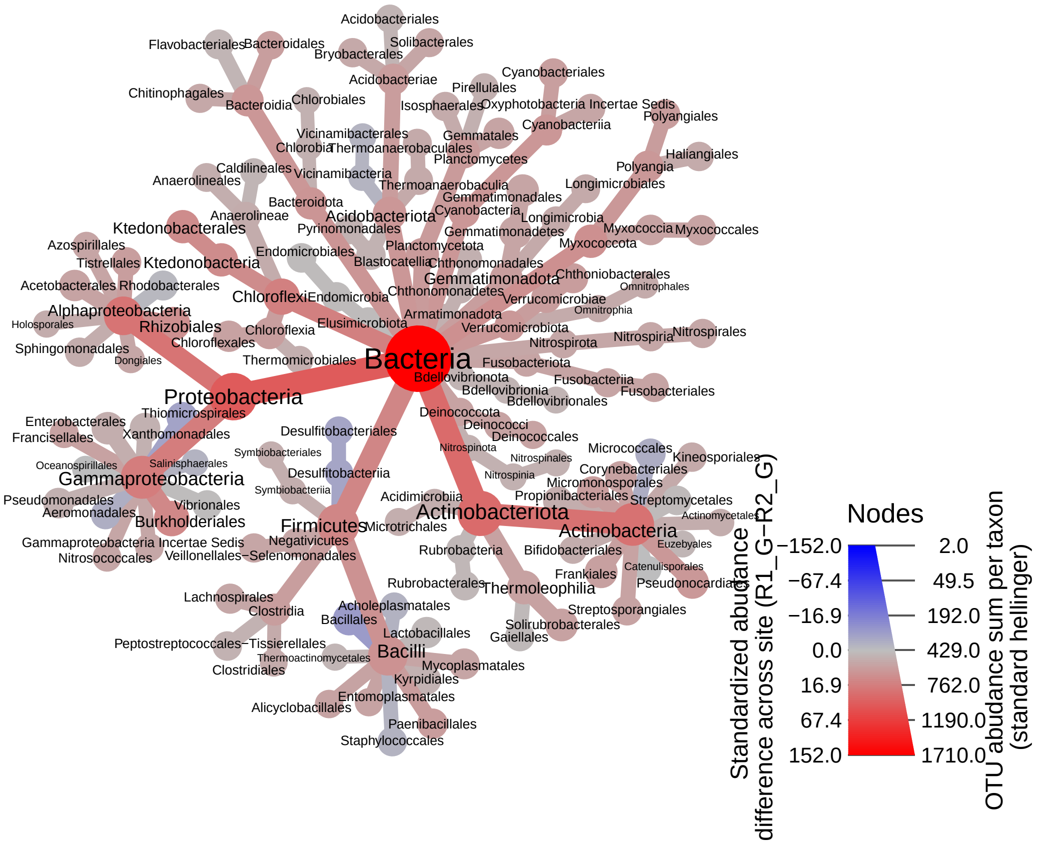
Here we see that in the phylum *Protobacteia* the class *Gammaprotobacteria* is mostly detected by R1 and *Alphaprotobacteria* is mostly detected by R1. *Fimicutes* and *Actinobacteria* are better detected by R1

Metagenomic R1 vs R2

```
meta_g_b_decond_table_rare%>%
  make.tax.tree.comp(method = c("R1_G","R2_G")) -> obj2
```

```
## Summing per-taxon counts from 158 columns in 2 groups for 616 taxa
```

```
obj2%>%
  filter_taxa(taxon_ranks == "o", supertaxa = TRUE)%>%
  heat_tree(node_label = gsub(pattern = "\\[|\\]", replacement = "", taxon_names),
            node_size = R1_G+R2_G,
            node_color = R1_G-R2_G,
            node_color_range = c("blue", "gray", "red"),
            node_color_interval = c(-max(abs(R1_G-R2_G)), max(abs(R1_G-R2_G))),
            node_color_axis_label = "Standardized abudance \n difference across site (R1_G-R2_G)",
            node_size_axis_label = "OTU abudance sum per taxon\n (standard hellinger)",
            layout = "davidson-harel", initial_layout = "reingold-tilford")
```



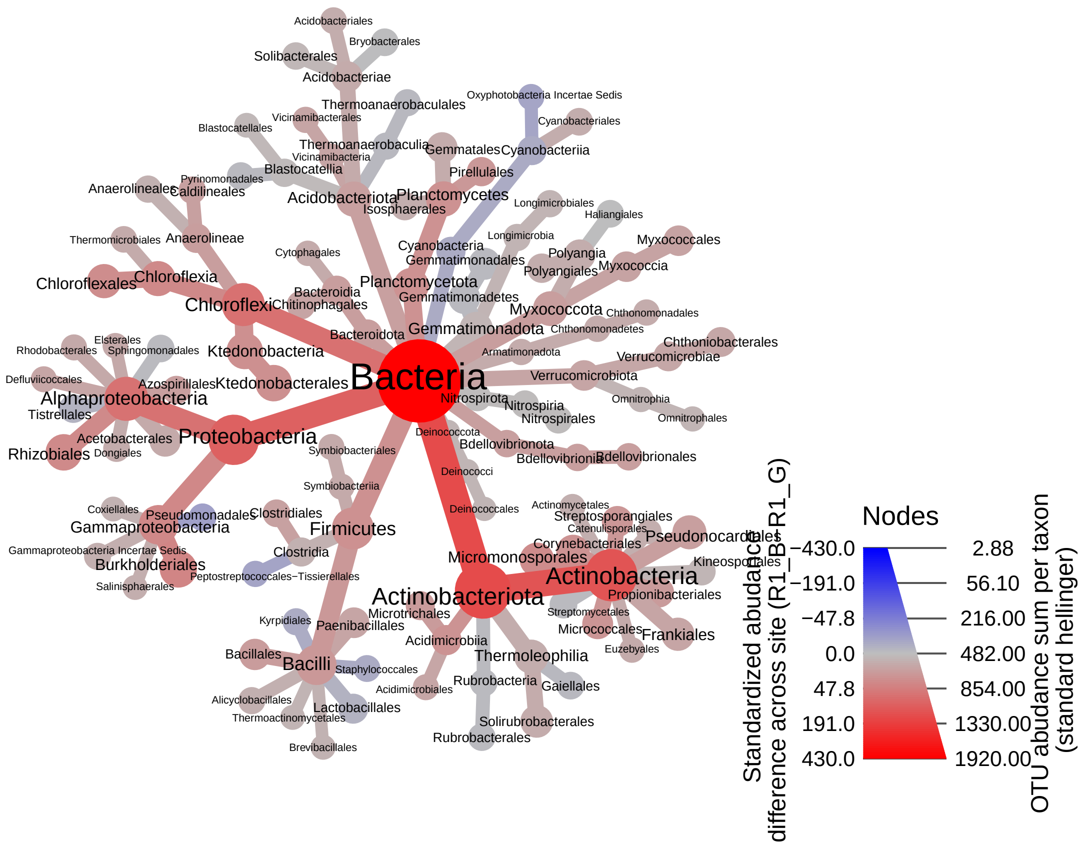
Different pattern for metaG than metaB here in phylum *Protobacteia* the class *Gammaprotobacteria* and *Alphaprotobacteria* are both more detected by R1. *Fimicutes* and *Actinobacteria* are also better detected by R1

Metabarcoding R1 vs Metagenomic R1

```
meta_g_b_decond_table_rare%>%
  make.tax.tree.comp(method = c("R1_B","R1_G")) -> obj3
```

```
## Summing per-taxon counts from 158 columns in 2 groups for 578 taxa
```

```
obj3%>%
  filter_taxa(taxon_ranks == "o", supertaxa = TRUE)%>%
  heat_tree(node_label = gsub(pattern = "\\[|\\]", replacement = "", taxon_names),
            node_size = R1_B+R1_G,
            node_color = R1_B-R1_G,
            node_color_range = c("blue", "gray", "red"),
            node_color_interval = c(-max(abs(R1_B - R1_G)), max(abs(R1_B - R1_G))),
            node_color_axis_label = "Standardized abudance \n difference across site (R1_B-R1_G)",
            node_size_axis_label = "OTU abudance sum per taxon\n (standard hellinger)",
            layout = "davidson-harel", initial_layout = "reingold-tilford")
```

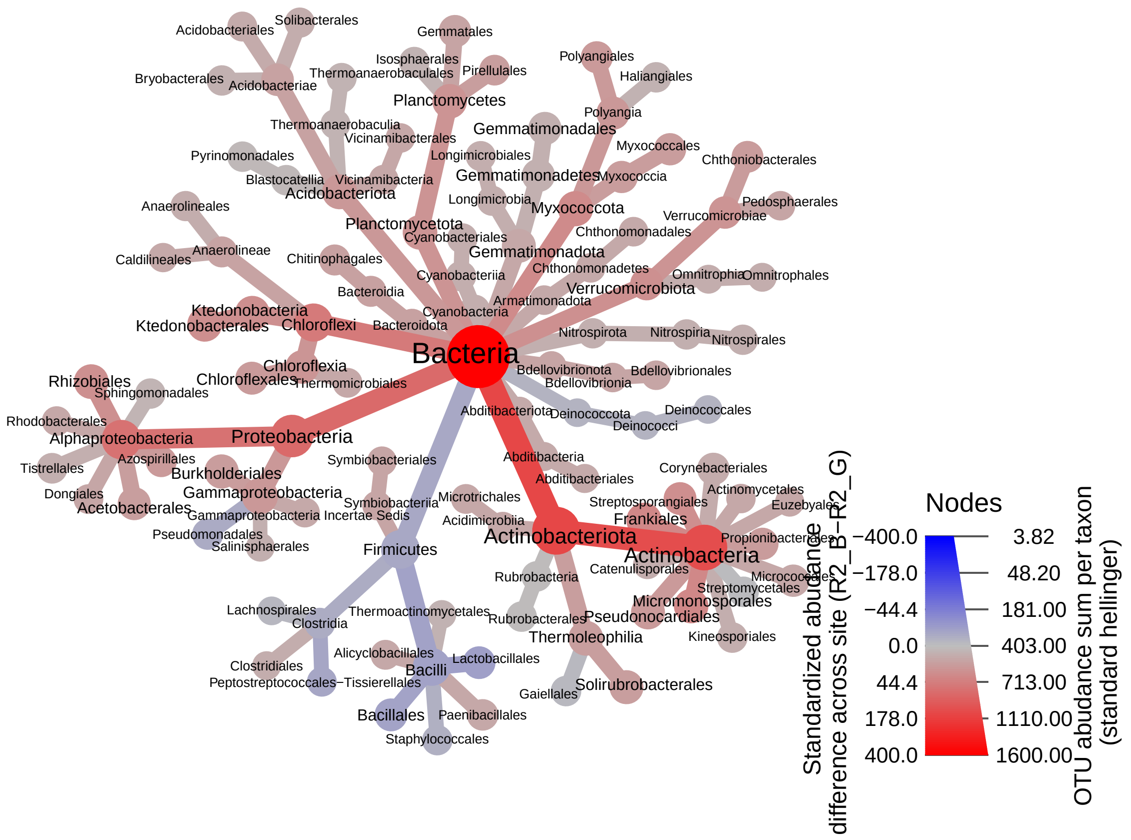


Metabarcoding R2 vs Metagenomic R2

```
meta_g_b_decond_table_rare%>%
  make.tax.tree.comp(method = c("R2_B","R2_G")) -> obj4
```

```
## Summing per-taxon counts from 158 columns in 2 groups for 505 taxa
```

```
obj4%>%
  filter_taxa(taxon_ranks == "o", supertaxa = TRUE)%>%
  heat_tree(node_label = gsub(pattern = "\\[|\\]", replacement = "", taxon_names),
            node_size = R2_B+R2_G,
            node_color = R2_B-R2_G,
            node_color_range = c("blue", "gray", "red"),
            node_color_interval = c(-max(abs(R2_B-R2_G)), max(abs(R2_B-R2_G))),
            node_color_axis_label = "Standardized abudance \n difference across site (R2_B-R2_G)",
            node_size_axis_label = "OTU abudance sum per taxon\n (standard hellinger)",
            layout = "davidson-harel", initial_layout = "reingold-tilford")
```


Acidobacteriales
Solibacterales
Gemmatales
Isosphaerales
Pirellulales
Polyangiales
Haliangiales
Bryobacterales
Acidobacteriae
Thermoanaerobaculales
Planctomycetes
Polyangia
Thermoanaerobaculia
Vicinamibacterales
Gemmatimonadales
Myxococcales
Chthoniobacterales
Pyrinomonadales
Longimicrobiales
Blastocatellia
Vicinamibacteria
Gemmatimonadetes
Myxococcia
Acidobacteriota
Longimicrobia
Myxococcota
Verrucomicrobiae
Pedosphaerales
Anaerolineales
Planctomycetota
Cyanobacteriales
Chthonomonadales
Gemmatimonadota
Anaerolineae
Caldilineales
Chitinophagales
Chthonomonadetes
Omnitrophia
Omnitrophales
Cyanobacteriia
Verrucomicrobiota
Bacteroidia
Armatimonadota
Ktedonobacteria
Cyanobacteria
Ktedonobacterales
Chloroflexi
Bacteroidota
Nitrospirota
Nitrospiria
Nitrospirales
Bacteria
Chloroflexia
Bdellovibrionota
Bdellovibrionales
Rhizobiales
Chloroflexales
Thermomicrobiales
Bdellovibrionia
Sphingomonadales
Deinococcales
Abditibacteriota
Deinococcota
Rhodobacterales
Deinococci
Alphaproteobacteria
Proteobacteria
Azospirillales
Abditibacteria
Symbiobacteriales
Corynebacteriales
Tistrellales
Burkholderiales
Abditibacteriales
Actinomycetales
Gammaproteobacteria
Dongiales
Microtrichales
Streptosporangiales
Euzebyales
Acetobacterales
Symbiobacteriia
Frankiales
Gammaproteobacteria Incertae Sedis
Acidimicrobiia
Pseudomonadales
Actinobacteriota
Propionibacteriales
Salinisphaerales
Actinobacteria
Firmicutes
Catenulisporales
Micrococcales
Rubrobacteria
Streptomycetales
Micromonosporales
Lachnospirales
Thermoactinomycetales
Rubrobacterales
Pseudonocardiales
Clostridia
Thermoleophilia
Kineosporiales
Alicyclobacillales
Lactobacillales
Clostridiales
Bacilli
Peptostreptococcales−Tissierellales
Solirubrobacterales
Gaiellales
Bacillales
Paenibacillales
Staphylococcales
Nodes
Standardized abudance difference across site (R2_B−R2_G)
−400.0
−178.0
−44.4
0.0
44.4
178.0
400.0
OTU abudance sum per taxon (standard hellinger)
3.82
48.20
181.00
403.00
713.00
1110.00
1600.00