

Figures

Sara Beier

6/9/2022

Contents

1 Packages	1
2 Functions	2
3 Load and format input data	3
4 Scatterplot RRN_IMG versus RRN_rrnDB and within genus variance of RRN_rrnDB (Figure S1)	5
5 Scatterplot RRN_IMG, RRN_rrnDB and IMG genome size against the corresponding read based values (Figure S2)	6
6 PCAs (traits averaged at the species level)	8
6.1 PCA including 9 genomic traits as variables (Figure 3)	8
6.2 Contribution to total variability of all principal components (Figure S2)	11
6.3 PCAs after random removal of 1,2,3 or 4 of the variables (Figure S3)	12
7 Pairwise correlations (traits averaged at the species level)	15
7.1 Pairwise scatterplots among 9 genomic traits (Figure 4)	15
7.2 Pairwise scatterplots of resistance related traits versus CUB parameters (Figure 5)	16
8 Partial correlations (Table 1)	19
9 Habitat specific PCAs (traits considered at the genome level, Figure 6)	20
10 Mantel correlograms (Figure 7)	25

This file includes the R code used to create Figures in Beier et al, doi: <https://doi.org/10.1101/2021.07.23.453341>. The input files are either online available or were created as detailed in scripts_genomic.traits.md and scripts_phylogenetic.signal.md.

1 Packages

```
rm(list = ls())
library(tidyverse)
library(vegan)
library("PerformanceAnalytics")
```

2 Functions

```
# panel.smooth.mod
## The function panel.smooth.mod is delineated from the
## function panel.smooth [graphics] with the setting 'lwd=3' added
panel.smooth.mod <- function (x, y, col = par("col"), bg = NA, pch = par("pch"),
                               cex = 1, col.smooth = 2, span = 2/3, iter = 3, ...)
{
  points(x, y, pch = pch, col = col, bg = bg, cex = cex)
  ok <- is.finite(x) & is.finite(y)
  if (any(ok))
    lines(stats::lowess(x[ok], y[ok], f = span, iter = iter),
          col = col.smooth, lwd=3, ...) #add lwd=3
}

# chart.Correlation.mod
## The function chart.Correlation.mod is delineated from the
## function chart.Correlation [PerformanceAnalytics]
## with some graphical modifications.
## Lines in the original chart.Correlation that were ignored
## were deactivated with '#'.
## Comments indicate lines that were added or changed.
chart.Correlation.mod <-function (R, histogram = TRUE, method = c("pearson", "kendall",
                                                               "spearman"), ...)
{
  x = checkData(R, method = "matrix")
  if (missing(method))
    method = method[1]
  cormeth <- method
  panel.cor <- function(x, y, digits = 2, prefix = "", use = "pairwise.complete.obs",
                        method = cormeth, cex.cor, col='red',...) {
    usr <- par("usr")
    on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- cor(x, y, use = use, method = method)
    txt <- format(c(r, 0.123456789), digits = digits)[1]
    txt <- paste(prefix, txt, sep = "")
    if (missing(cex.cor))
      cex <- 0.8/strwidth(txt)
    test <- cor.test(as.numeric(x), as.numeric(y), method = method)
    #Signif <- symnum(test$p.value, corr = FALSE, na = FALSE,
    #                  cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
    #                  symbols = c("***", "**", "*", ".", " "))
    text(0.5, 0.5, txt, cex = cex * (abs(r) + 0.3)/1.3)
    #text(0.8, 0.8, Signif, cex = cex, col = 2)
  }
  f <- function(t) {
    dnorm(t, mean = mean(x), sd = sd.xts(x))
  }
  dotargs <- list(...)
  dotargs$method <- NULL
  rm(method)
  hist.panel = function(x, ... = NULL) {
    par(new = TRUE)
```

```

#hist(x, col = "light gray", probability = TRUE, axes = FALSE,
#      main = "", breaks = "FD", col.main="red")

#added line: replace original histogram
hist(x, col = "light gray", probability = TRUE, axes = FALSE,
      main = "", breaks = "FD")
#lines(density(x, na.rm = TRUE), col = "red", lwd = 1)
#rug(x)
}
if (histogram)
  pairs(x, gap = 0, lower.panel = panel.smooth.mod, upper.panel = panel.cor,
        diag.panel = hist.panel, cex.axis=2) #add cex.axis
#use panel.smooth.mod function with bolder trend line
else pairs(x, gap = 0, lower.panel = panel.smooth.mod, upper.panel = panel.cor)
}

#unsigned.range
## The unsigned.range determines graphical limits (xlim, ylim) for PCA biplots
unsigned.range <- function(x) c(-abs(min(x, na.rm = TRUE)), abs(max(x, na.rm = TRUE)))

```

3 Load and format input data

Input files are Table S1, RRN values available via the Ribosomal RNA Operon Copy Number Database (rrnDB) as well as mantel correlogram output files created as detailed in scripts_phylogenetic.signal.md

```

# Load habitat information, GTDB taxonomy and trait values from Table S1
gtraits <- tibble(read.table ("TableS1.tsv", header=T, sep='\t', fill=T)) %>%
  select(IMG.Genome.ID,Habitat,
         GTDB_division,GTDB_phylum,GTDB_class,GTDB_order,GTDB_family,GTDB_genus,
         GTDB_species, FASTANI_species,
         Genome.size,X.GC,RRN_IMG,RRN_rrnDB,X.HGT,CUB.F.,Generation.time..Vieira.Silva.,
         Generation.time..gRodon.,Gene.duplication,Gene.richness,X.TF,Prophages)%>%
  #Genome.size values displayed as Mbp
  mutate(Genome.size= Genome.size/1000000) %>% #

  #create column with %HGT values larger 65% removed
  mutate(per.HGT.corr = ifelse(X.HGT < 65, X.HGT, NA)) %>% #

  #create column with Prophages >20 removed
  mutate(Prophages.corr = ifelse(Prophages < 20, Prophages, NA)) #
gtraits

## # A tibble: 17,856 x 24
##   IMG.Genome.ID Habitat      GTDB_division GTDB_phylum GTDB_class GTDB_order
##   <dbl> <chr>          <chr>           <chr>       <chr>       <chr>
## 1 2740891980 ""            Archaea        Altarchaeo~ Altarchae~ IMC4
## 2 2740891986 ""            Archaea        Asgardarch~ Lokiarcha~ Odinarcha~
## 3 2740891981 ""            Archaea        Asgardarch~ Thorarcha~ Thorarcha~
## 4 2528311132 "Aquatic, Mari~ Archaea       Halobacter~ Archaeogl~ Archaeogl~
## 5 2522125074 "Aquatic, Mari~ Archaea       Halobacter~ Archaeogl~ Archaeogl~
## 6 646311906 "Aquatic, Deep~ Archaea       Halobacter~ Archaeogl~ Archaeogl~
## 7 2504136002 "Aquatic, Deep~ Archaea       Halobacter~ Archaeogl~ Archaeogl~

```

```

## 8      646564534 "Aquatic, Fres~ Archaea      Halobacter~ Archaeogl~ Archaeogl~
## 9      2634166507 ""                      Archaea      Halobacter~ Archaeogl~ Archaeogl~
## 10     2728369302 ""                      Archaea      Halobacter~ Archaeogl~ JdFR-21
## # ... with 17,846 more rows, and 18 more variables: GTDB_family <chr>,
## #   GTDB_genus <chr>, GTDB_species <chr>, FASTANI_species <chr>,
## #   Genome.size <dbl>, X.GC <dbl>, RRN_IMG <int>, RRN_rrnDB <dbl>, X.HGT <dbl>,
## #   CUB.F. <dbl>, Generation.time..Vieira.Silva. <dbl>,
## #   Generation.time..gRodon. <dbl>, Gene.duplication <dbl>,
## #   Gene.richness <int>, X.TF <dbl>, Prophages <int>, per.HGT.corr <dbl>,
## #   Prophages.corr <int>

# Load rrnDB RRN trait data
rrnDB <- tibble(read.csv ("rrnDB-5.7.tsv", sep='\t', header=T, fill=T)[,c(1,7,12)])
rrnDB

```

```

## # A tibble: 20,642 x 3
##   Data.source.record.id RDP.taxa X16S.gene.count
##   <chr>                <chr>           <int>
## 1 rrnDBv3-1403          ""                 4
## 2 rrnDBv3-1404          ""                 4
## 3 rrnDBv3-1405          ""                 4
## 4 rrnDBv3-1407          ""                 2
## 5 rrnDBv3-1408          ""                 2
## 6 rrnDBv3-1409          ""                 NA
## 7 rrnDBv3-1410          ""                 7
## 8 rrnDBv3-1411          ""                 7
## 9 rrnDBv3-1412          ""                 7
## 10 rrnDBv3-1413         ""                 7
## # ... with 20,632 more rows

```

```

# Load read based RRN trait data
rrn_reads <- tibble(read.table ("TableS2.tsv", header=T, sep='\t', fill=T))
rrn_reads

```

```

## # A tibble: 1,572 x 9
##   IMG.Genome.ID runID      RRN_IMG RRN_rrnDB RRN_read.based Genome.size
##   <dbl> <chr>      <int>    <dbl>       <int>       <int>
## 1 2517093035 DRR001004      4     4.32        5     2579188
## 2 2602041654 DRR002764      1      NA         6     3912387
## 3 2602041661 DRR013904      1     4.96        9     5064759
## 4 2609460248 DRR018584      1      NA         3     3085726
## 5 2582581020 DRR021662      1     7.64        7     2129279
## 6 2639762744 DRR024543      5     10.4       13     6001163
## 7 2690315807 DRR033868      1     10.6       15     6759522
## 8 2651869759 DRR037147      1     10.4       16     5792751
## 9 2551306054 DRR040583      1      NA         6     3148763
## 10 2740892220 DRR045531     1     3.96        5     2209887
## # ... with 1,562 more rows, and 3 more variables: Genome.size_read.based <dbl>,
## #   Genome.equivalents_read.based <dbl>, GTDB_genus <chr>

```

```

# Load output data from mantel correlogram analyses (scripts_phylogenetic.signal.md)
mpm.2 <- read.table("mpm.2.tab", sep= '\t', header=T)
mpm.3 <- read.table("mpm.3.tab", sep= '\t', header=T)
mpm.4 <- read.table("mpm.rrn.tab", sep= '\t', header=T)
mpm.5 <- read.table("mpm.5.tab", sep= '\t', header=T)
mpm.6 <- read.table("mpm.6.tab", sep= '\t', header=T)

```

```

mpm.7 <- read.table("mpm.7.tab", sep= '\t', header=T)
mpm.8 <- read.table("mpm.8.tab", sep= '\t', header=T)
mpm.9 <- read.table("mpm.9.tab", sep= '\t', header=T)
mpm.10 <- read.table("mpm.10.tab", sep= '\t', header=T)
mpm.11<- read.table("mpm.11.tab", sep= '\t', header=T)

```

4 Scatterplot RRN_IMG versus RRN_rrnDB and within genus variance of RRN_rrnDB (Figure S1)

```

# Prepare data set for RRN_IMG versus RRN_rrnDB scatterplot
dat <- gtraits %>%
  filter(RRN_IMG<20) %>% #remove rows with RRN_IMG < 20
  filter(!is.na(RRN_IMG) & !is.na(RRN_rrnDB)) #remove rows with NA in either RRN_IMG or RRN_rrnDB
dat

## # A tibble: 12,870 x 24
##   IMG.Genome.ID Habitat      GTDB_division GTDB_phylum GTDB_class GTDB_order
##   <dbl> <chr>          <chr>           <chr>          <chr>          <chr>
## 1 2528311132 "Aquatic, Mari~ Archaea    Halobacter~ Archaeogl~ Archaeogl~
## 2 2522125074 "Aquatic, Mari~ Archaea    Halobacter~ Archaeogl~ Archaeogl~
## 3 646311906 "Aquatic, Deep~ Archaea    Halobacter~ Archaeogl~ Archaeogl~
## 4 2504136002 "Aquatic, Deep~ Archaea    Halobacter~ Archaeogl~ Archaeogl~
## 5 646564534 "Aquatic, Fres~ Archaea    Halobacter~ Archaeogl~ Archaeogl~
## 6 2634166507 ""             Archaea    Halobacter~ Archaeogl~ Archaeogl~
## 7 648028029 "Fermented foo~ Archaea    Halobacter~ Halobacte~ Halobacte~
## 8 2528311097 "Fermented foo~ Archaea    Halobacter~ Halobacte~ Halobacte~
## 9 2684622629 ""             Archaea    Halobacter~ Halobacte~ Halobacte~
## 10 2630968753 ""            Archaea    Halobacter~ Halobacte~ Halobacte~
## # ... with 12,860 more rows, and 18 more variables: GTDB_family <chr>,
## #   GTDB_genus <chr>, GTDB_species <chr>, FASTANI_species <chr>,
## #   Genome.size <dbl>, X.GC <dbl>, RRN_IMG <int>, RRN_rrnDB <dbl>, X.HGT <dbl>,
## #   CUB.F. <dbl>, Generation.time..Vieira.Silva. <dbl>,
## #   Generation.time..gRodon. <dbl>, Gene.duplication <dbl>,
## #   Gene.richness <int>, X.TF <dbl>, Prophages <int>, per.HGT.corr <dbl>,
## #   Prophages.corr <int>

# Create data set containing within genus sd of rrnDB RRN values
rrnDB.genus.sd <- tibble(aggregate(. ~ RDP.taxa, data=rrnDB[grep("(genus)", rrnDB$RDP.taxa),
c(2:3)], FUN=sd))

rrnDB.genus.sd

## # A tibble: 1,199 x 2
##   RDP.taxa                  X16S.gene.count
##   <chr>                      <dbl>
## 1 "\Candidatus Korarchaeum\" (genus)        NA
## 2 "Abiotrophia (genus)"         NA
## 3 "Acetoanaerobium (genus)"     NA
## 4 "Acetobacter (genus)"        0.575
## 5 "Acetobacterium (genus)"     NA
## 6 "Acetohalobium (genus)"      NA
## 7 "Acetothermia_genera_incertae_sedis (genus)" NA
## 8 "Acholeplasma (genus)"       0.599
## 9 "Achromobacter (genus)"      0.502

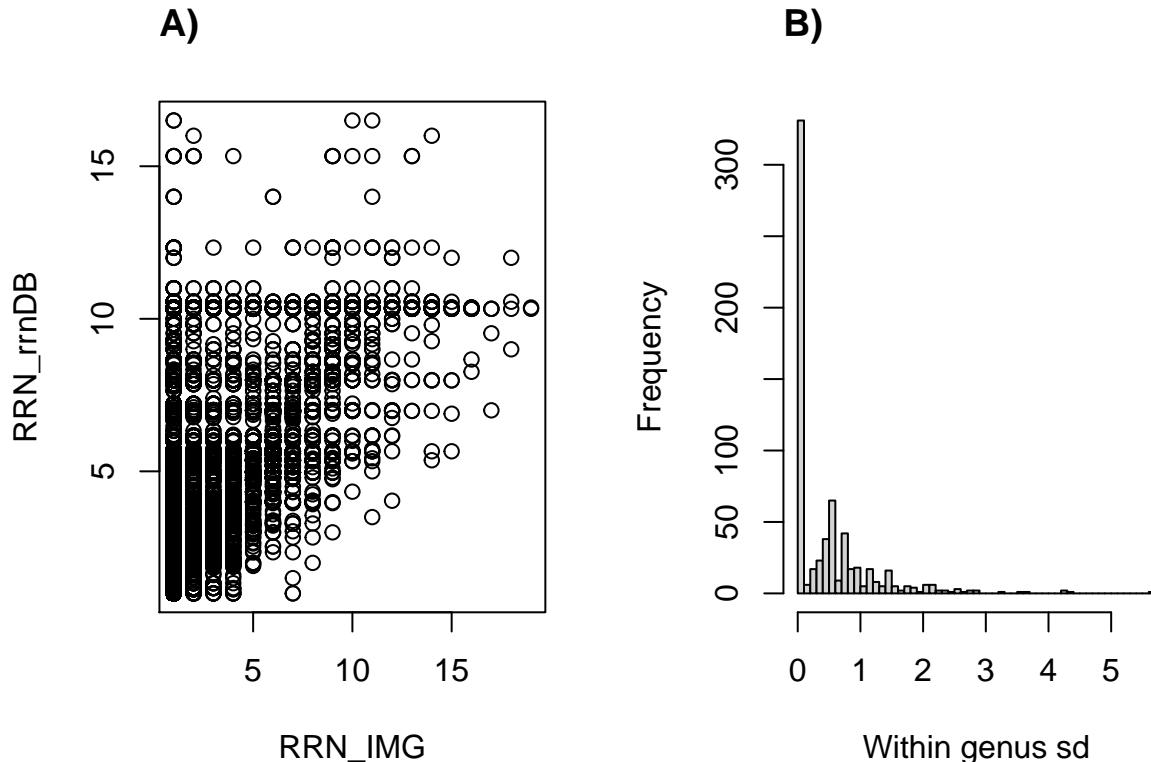
```

```

## 10 "Acidaminococcus (genus)"                                2.12
## # ... with 1,189 more rows

# Figure S1
par(mfrow=c(1,2))
plot(dat$RRN_IMG, dat$RRN_rrnDB, xlab='RRN_IMG', ylab='RRN_rrnDB')
title('A)', adj=0)
hist(rrnDB.genus.sd$X16S.gene.count, breaks=50, xlab="Within genus sd", main="")
title('B)', adj=0)

```



```

## pdf
## 2

```

Figure S1 A) The scatter plot of RRN_IMG versus RRN_rrnDB indicates that RRN values available via the JGI/IMG platform are often biased towards an underestimation of RRN counts that are available via the rrnDB. B) The histogram of withing genus standard deviations (sd) of RRN values from the rrnDB demonstrates a large majority of genera with $sd < 1$.

5 Scatterplot RRN_IMG, RRN_rrnDB and IMG genome size against the corresponding read based values (Figure S2)

```

# Pearson correlation
c1 <- cor.test(rrn_reads$RRN_IMG, rrrn_reads$RRN_read_based)
c2 <- cor.test(rrn_reads$RRN_rrnDB, rrrn_reads$RRN_read_based)
c3 <- cor.test(rrn_reads$Genome.size, rrrn_reads$Genome.size_read_based)

```

```

# Figure S2
par(mfrow = c(1, 3))
par(mgp = c(1.5, 0.4, 0))

plot(rrn_reads$RRN_IMG, rrn_reads$RRN_read_based, xlab = "RRN_IMG", ylab = "RRN_read_based")
title("A)", adj = 0, line = 0.7)
legend("topright", paste("r=", round(c1[[4]], 2), " p=", round(c1[[3]], 2)), bty = "n")

plot(rrn_reads[, 4, drop = T], rrn_reads[, 5, drop = T], xlab = "RRN_rrnDB",
      ylab = "RRN_read_based")
title("B)", adj = 0, line = 0.7)
legend("topright", paste("r=", round(c2[[4]], 2), " p=", round(c2[[3]], 3)), bty = "n")

plot(rrn_reads$Genome.size, rrn_reads$Genome.size_read_based, xlab = "Genome.size_IMG",
      ylab = "Genome.size_read_based")
title("C)", adj = 0, line = 0.7)
legend("topright", paste("r=", round(c3[[4]], 2), " p=", round(c3[[3]], 3)), bty = "n")

```

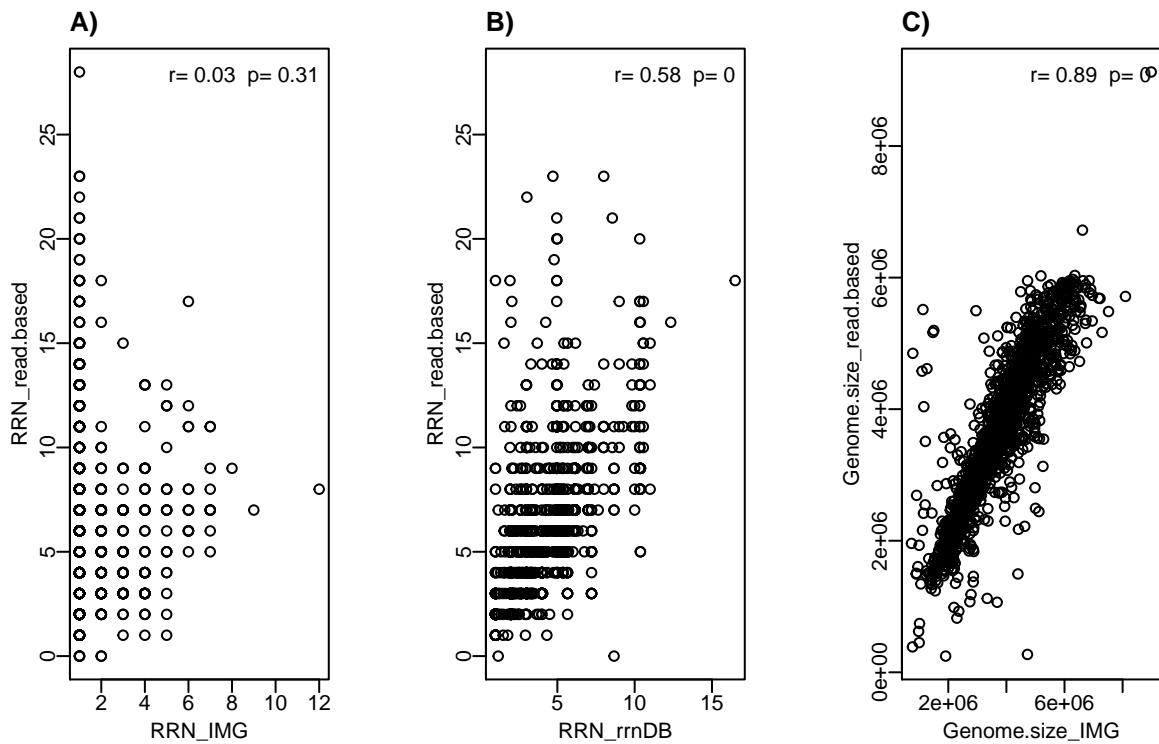


Figure S3: . A) Pearson correlation between the read based RRN estimate and the JGI RRN estimate. B) Pearson correlation between the read based RRN estimate and the rrnDB RRN estimate as given in Table S1. C) Pearson correlation between the read based genome size estimate and the genome size given in the JGI database.

```
## pdf
```

```
##    2
```

6 PCAs (traits averaged at the species level)

6.1 PCA including 9 genomic traits as variables (Figure 3)

Trait data were averaged at the species level via the mean function to remove phylogenetic redundancy in our data set (e.g. due to numerous close relatives affiliating with individual clinically relevant strains)

```
#Aggregate trait table at the species level
gtraits.sp <- gtraits %>%
  select(GTDB_division, GTDB_phylum, GTDB_class, GTDB_order, GTDB_family, GTDB_genus,
         GTDB_species, FASTANI_species,
         Genome.size, X.TF, Gene.duplication, Gene.richness, CUB.F., RRN_rrnDB, Prophages.corr,
         per.HGT.corr, X.GC, Generation.time..Vieira.Silva., Generation.time..gRodon.) %>%
  group_by(GTDB_division, GTDB_phylum, GTDB_class, GTDB_order, GTDB_family, GTDB_genus,
           GTDB_species, FASTANI_species) %>%
  summarise_at(vars(Genome.size:Generation.time..gRodon.), mean, na.rm = TRUE) %>%
  ungroup %>%
  mutate(log.HGT_per = log(per.HGT.corr+0.01)) %>% #log(x+0.01) data transformation
  mutate(log.d = log(Generation.time..Vieira.Silva.)) %>% #log(x) data transformation
  mutate(log.d.gRodon = log(Generation.time..gRodon.)) #log(x) data transformation
gtraits.sp

## # A tibble: 8,847 x 22
##   GTDB_division GTDB_phylum     GTDB_class     GTDB_order GTDB_family GTDB_genus
##   <chr>          <chr>          <chr>          <chr>      <chr>      <chr>
## 1 Archaea       Altarchaeota  Altarchaeia   IMC4        WOR-SM1-SCG WOR-SM1-S~
## 2 Archaea       Asgardarchaeota Lokiarchaeia~ Odinarcha~ LCB-4       LCB-4
## 3 Archaea       Asgardarchaeota Thorarchaeia Thorarcha~ Thorarchae~ SMTZ1-83
## 4 Archaea       Halobacteriota Archaeoglobi Archaeogl~ Archaeogl~ Archaeogl~
## 5 Archaea       Halobacteriota Archaeoglobi Archaeogl~ Archaeogl~ Archaeogl~
## 6 Archaea       Halobacteriota Archaeoglobi Archaeogl~ Archaeogl~ Archaeogl~
## 7 Archaea       Halobacteriota Archaeoglobi Archaeogl~ Archaeogl~ Archaeogl~
## 8 Archaea       Halobacteriota Archaeoglobi Archaeogl~ Archaeogl~ Ferroglob~
## 9 Archaea       Halobacteriota Archaeoglobi Archaeogl~ Archaeogl~ Geoglobus
## 10 Archaea      Halobacteriota Archaeoglobi JdFR-21    JdFR-21    JdFR-21
## # ... with 8,837 more rows, and 16 more variables: GTDB_species <chr>,
## #   FASTANI_species <chr>, Genome.size <dbl>, X.TF <dbl>,
## #   Gene.duplication <dbl>, Gene.richness <dbl>, CUB.F. <dbl>, RRN_rrnDB <dbl>,
## #   Prophages.corr <dbl>, per.HGT.corr <dbl>, X.GC <dbl>,
## #   Generation.time..Vieira.Silva. <dbl>, Generation.time..gRodon. <dbl>,
## #   log.HGT_per <dbl>, log.d <dbl>, log.d.gRodon <dbl>
print(paste("#Nr phyla: ", length(unique(gtraits.sp$GTDB_phylum))))
```



```
## [1] "#Nr phyla: 73"
print(paste("#Nr classes: ", length(unique(gtraits.sp$GTDB_class))))
```



```
## [1] "#Nr classes: 172"
print(paste("#Nr orders: ", length(unique(gtraits.sp$GTDB_order))))
```



```
## [1] "#Nr orders: 382"
```

```

print(paste("#Nr families: ", length(unique(gtraits.sp$GTDB_family)))))

## [1] "#Nr families: 762"

print(paste("#Nr genera: ", length(unique(gtraits.sp$GTDB_genus)))))

## [1] "#Nr genera: 2669"

print(paste("#Nr species: ", dim(gtraits.sp)[1])))

## [1] "#Nr species: 8847"

# Select columns relevant for Figure 3 and Figure 4
vars <- gtraits.sp %>%
  select(Genome.size,X.TF,Gene.duplication,Gene.richness,CUB.F.,RRN_rrnDB,Prophages.corr,
         log.HGT_per,X.GC)

# Change column names for Figure 3 and Figure 4
colnames(vars) <- c('Genome size', '%TF', 'Gene duplication', 'Gene richness',
                     'CUB (F)', 'RRN', 'Prophages', '%HGT (log)', '%GC')

# Principle component analysis
pca<-prcomp(na.omit(vars),center=TRUE, scale=TRUE)

# Proportion variance explained in component 1 and 2
prop.1<-summary(pca)$importance[2]*100
prop.2<-summary(pca)$importance[5]*100

# Extract the eigenvectors and the loading factors
pca.scores<-pca$x[,1:2]
pca.loadings<-pca$rotation[,1:2]

# Normalize PC axes to unit length
pca.scores[,1]<-pca.scores[,1]*(1/sqrt(sum(pca.scores[,1]^2)))
pca.scores[,2]<-pca.scores[,2]*(1/sqrt(sum(pca.scores[,2]^2)))

# Set scaling parameters for biplot
sc<-1.5
xlim <- unsigned.range(pca.scores[,1])*sc
ylim <- unsigned.range(pca.scores[,1])*sc
ratio <-max(abs(unsigned.range(pca.loadings[,1]))[1],
            -unsigned.range(pca.loadings[,1])[2])/abs(xlim[1]-xlim[2]),
            abs(unsigned.range(pca.loadings[,2]))[1]
            -unsigned.range(pca.loadings[,2])[2])/abs(ylim[1]-ylim[2]))*sc/1.3

# Set colors
col=c(rep("black",4), rep("#E69F00",2), c("gray55","#E69F00","gray55"))

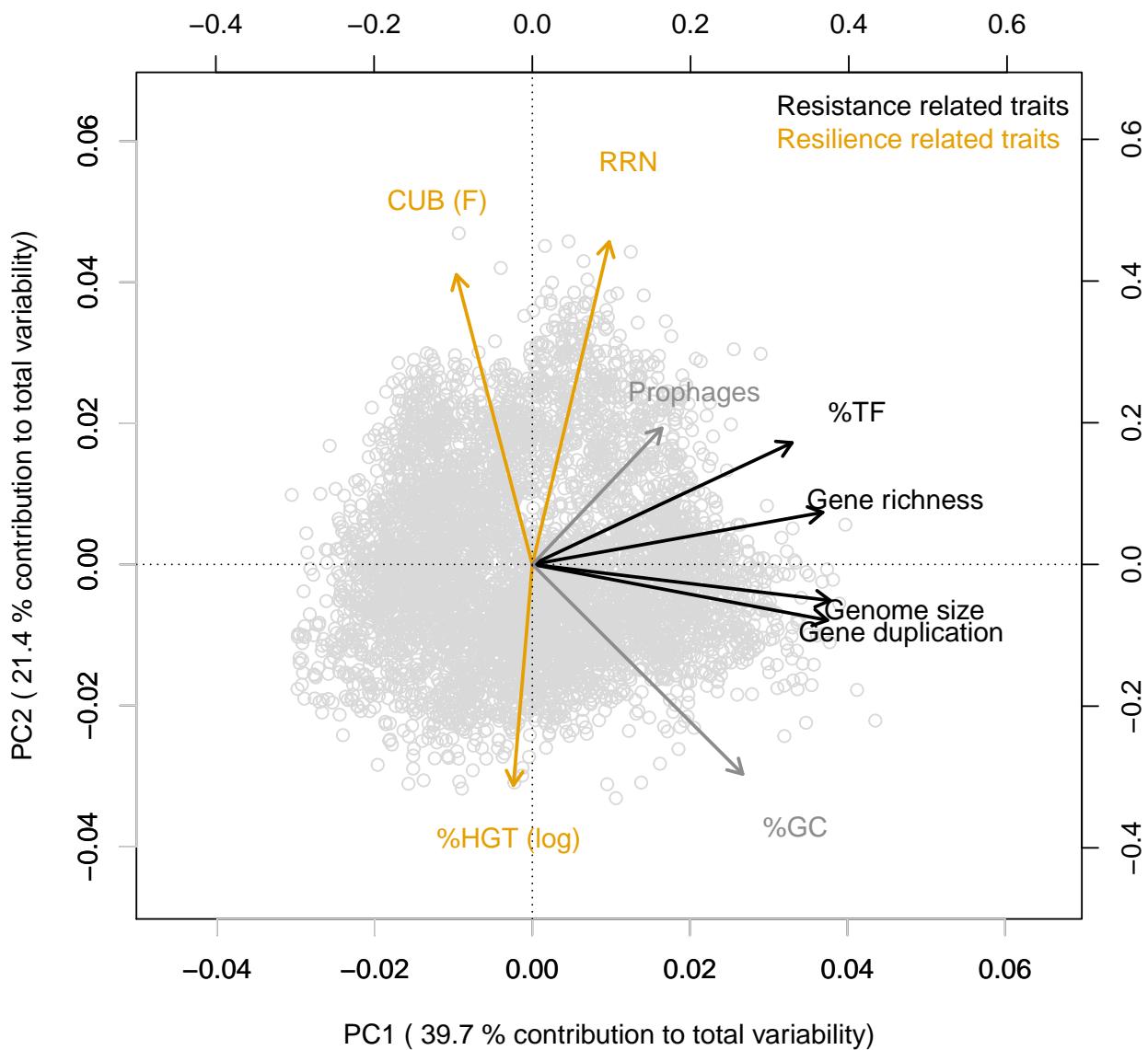
# PCA biplot
plot(pca.scores,xlim=xlim,ylim=ylim,
      ylab=paste("PC2 (",round(prop.2,1),"% contribution to total variability)"),
      xlab=paste("PC1 (",round(prop.1,1),"% contribution to total variability)"),
      col='gray85',cex=1)
axis(1, col = "gray85")
axis(2, col = "gray85")

```

```

par(new=TRUE)
plot(pca.loadings, axes=FALSE, type="n", xlim=xlim*ratio, ylim=ylim*ratio, xlab = "", ylab = "")
axis(3, col = "black")
axis(4, col = "black")
text(pca.loadings, labels = rownames(pca.loadings), cex = 1, col = col)
arrows(0, 0, pca.loadings[, 1] * 0.8, pca.loadings[, 2] * 0.8, col = col, length = 0.1, lwd = 2)
abline(v=0, h=0, lty="dotted")
legend('topright', c("Resistance related traits", "Resilience related traits"),
       text.col=c('black', '#E69F00'), bty="n")

```



```

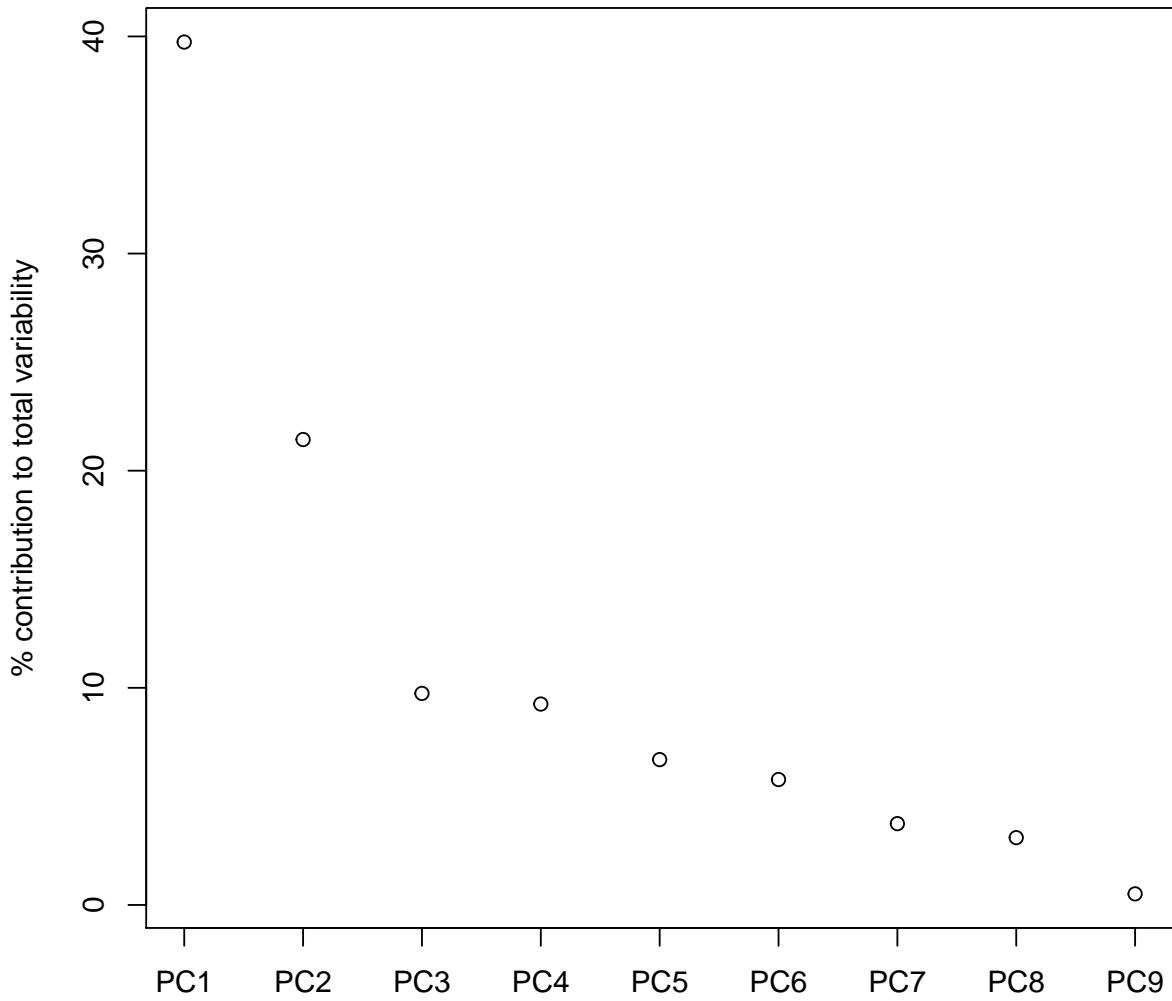
## pdf
## 2

```

Figure 3: Principal Component Analyses of trait values averaged at the species level. (The positions of some text labels were later adjusted with an external graphic program)

6.2 Contribution to total variability of all principal components (Figure S2)

```
# Figure S3
plot(summary(pca)$importance[c(2, 5, 8, 11, 14, 17, 20, 23, 26)] * 100,
      ylab = "% contribution to total variability", xlab = "", xaxt = "n")
axis(1, at = 1:9, labels = colnames(summary(pca)$importance))
```



```
## pdf
## 2
```

Figure S3: Percent contribution to total variability of the principal components PC1 to PC9

6.3 PCAs after random removal of 1,2,3 or for of the variables (Figure S3)

```

# Randomly remove 1,2,3,4 parameters

set.seed(2022)
vars8 <- sample(vars, 8)
vars7 <- sample(vars8, 7)
vars6 <- sample(vars7, 6)
vars5 <- sample(vars6, 5)

# Principal component analyses
pca8 <- prcomp(na.omit(vars8), center = TRUE, scale = TRUE)
pca7 <- prcomp(na.omit(vars7), center = TRUE, scale = TRUE)
pca6 <- prcomp(na.omit(vars6), center = TRUE, scale = TRUE)
pca5 <- prcomp(na.omit(vars5), center = TRUE, scale = TRUE)

# Extract the eigenvectors and the loading factors and normalize PC
# axes to unit length
pca8.scores <- pca8$x[, 1:2]
pca8.loadings <- pca8$rotation[, 1:2][order(row.names(pca8$rotation[, 1:2])), ]
pca8.scores[, 1] <- pca8.scores[, 1] * (1/sqrt(sum(pca8.scores[, 1]^2)))
pca8.scores[, 2] <- pca8.scores[, 2] * (1/sqrt(sum(pca8.scores[, 2]^2)))

pca7.scores <- pca7$x[, 1:2]
pca7.loadings <- pca7$rotation[, 1:2][order(row.names(pca7$rotation[, 1:2])), ]
pca7.scores[, 1] <- pca7.scores[, 1] * (1/sqrt(sum(pca7.scores[, 1]^2)))
pca7.scores[, 2] <- pca7.scores[, 2] * (1/sqrt(sum(pca7.scores[, 2]^2)))

pca6.scores <- pca6$x[, 1:2]
pca6.loadings <- pca6$rotation[, 1:2][order(row.names(pca6$rotation[, 1:2])), ]
pca6.scores[, 1] <- pca6.scores[, 1] * (1/sqrt(sum(pca6.scores[, 1]^2)))
pca6.scores[, 2] <- pca6.scores[, 2] * (1/sqrt(sum(pca6.scores[, 2]^2)))

pca5.scores <- pca5$x[, 1:2]
pca5.loadings <- pca5$rotation[, 1:2][order(row.names(pca5$rotation[, 1:2])), ]
pca5.scores[, 1] <- pca5.scores[, 1] * (1/sqrt(sum(pca5.scores[, 1]^2)))
pca5.scores[, 2] <- pca5.scores[, 2] * (1/sqrt(sum(pca5.scores[, 2]^2)))

# Set scaling parameters for biplots
sc <- 1.5
xlim8 <- unsigned.range(pca8.scores[, 1]) * sc
ylim8 <- unsigned.range(pca8.scores[, 2]) * sc
ratio <- max(abs(unsigned.range(pca8.loadings[, 1])[1] - unsigned.range(pca8.loadings[, 1])[2]))/abs(xlim8[1] - xlim8[2]), abs(unsigned.range(pca8.loadings[, 2])[1] - unsigned.range(pca8.loadings[, 2])[2]))/abs(ylim8[1] - ylim8[2])) *
sc/1.3

xlim7 <- unsigned.range(pca7.scores[, 1]) * sc
ylim7 <- unsigned.range(pca7.scores[, 2]) * sc
ratio <- max(abs(unsigned.range(pca7.loadings[, 1])[1] - unsigned.range(pca7.loadings[, 1])[2]))/abs(xlim7[1] - xlim7[2]), abs(unsigned.range(pca7.loadings[, 2])[1] - unsigned.range(pca7.loadings[, 2])[2]))/abs(ylim7[1] - ylim7[2])) *
sc/1.3

```

```

  1]) [2]) /abs(xlim7[1] - xlim7[2]), abs(unsigned.range(pca7.loadings[, 2])[1] - unsigned.range(pca7.loadings[, 2])[2]) /abs(ylim7[1] - ylim7[2])) * sc /1.3

xlim6 <- unsigned.range(pca6.scores[, 1]) * sc
ylim6 <- unsigned.range(pca6.scores[, 2]) * sc
ratio <- max(abs(unsigned.range(pca6.loadings[, 1])[1] - unsigned.range(pca6.loadings[, 1])[2]) /abs(xlim6[1] - xlim6[2]), abs(unsigned.range(pca6.loadings[, 2])[1] - unsigned.range(pca6.loadings[, 2])[2]) /abs(ylim6[1] - ylim6[2])) * sc /1.3

xlim5 <- unsigned.range(pca5.scores[, 1]) * sc
ylim5 <- unsigned.range(pca5.scores[, 2]) * sc
ratio <- max(abs(unsigned.range(pca5.loadings[, 1])[1] - unsigned.range(pca5.loadings[, 1])[2]) /abs(xlim5[1] - xlim5[2]), abs(unsigned.range(pca5.loadings[, 2])[1] - unsigned.range(pca5.loadings[, 2])[2]) /abs(ylim5[1] - ylim5[2])) * sc /1.3

# Colors
col = c(rep("black", 4), rep("#E69F00", 2), c("gray55", "#E69F00", "gray55"))
color.schema <- data.frame(names(vars), col)

# Figure S4
par(mfrow = c(2, 2))
col = merge(pca8.loadings, color.schema, by.x = 0, by.y = "names.vars.")[, 4]
plot(pca8.scores, xlim = xlim8, ylim = ylim8, col = "gray85", cex = 1)
par(new = TRUE)
plot(pca8.loadings, axes = FALSE, type = "n", xlim = xlim8 * ratio, ylim = ylim8 * ratio, xlab = "", ylab = "")
axis(3, col = "black")
axis(4, col = "black")
text(pca8.loadings, labels = rownames(pca8.loadings), cex = 1, col = col)
arrows(0, 0, pca8.loadings[, 1] * 0.8, pca8.loadings[, 2] * 0.8, col = col, length = 0.1, lwd = 2)
abline(v = 0, h = 0, lty = "dotted")
legend("topright", c("Resistance related traits", "Resilience related traits"), text.col = c("black", "#E69F00"), bty = "n")

col = merge(pca7.loadings, color.schema, by.x = 0, by.y = "names.vars.")[, 4]
plot(pca7.scores, xlim = xlim7, ylim = ylim7, col = "gray85", cex = 1)
par(new = TRUE)
plot(pca7.loadings, axes = FALSE, type = "n", xlim = xlim7 * ratio, ylim = ylim7 * ratio, xlab = "", ylab = "")
axis(3, col = "black")
axis(4, col = "black")
text(pca7.loadings, labels = rownames(pca7.loadings), cex = 1, col = col)
arrows(0, 0, pca7.loadings[, 1] * 0.7, pca7.loadings[, 2] * 0.8, col = col, length = 0.1, lwd = 2)
abline(v = 0, h = 0, lty = "dotted")

col = merge(pca6.loadings, color.schema, by.x = 0, by.y = "names.vars.")[, 4]

```

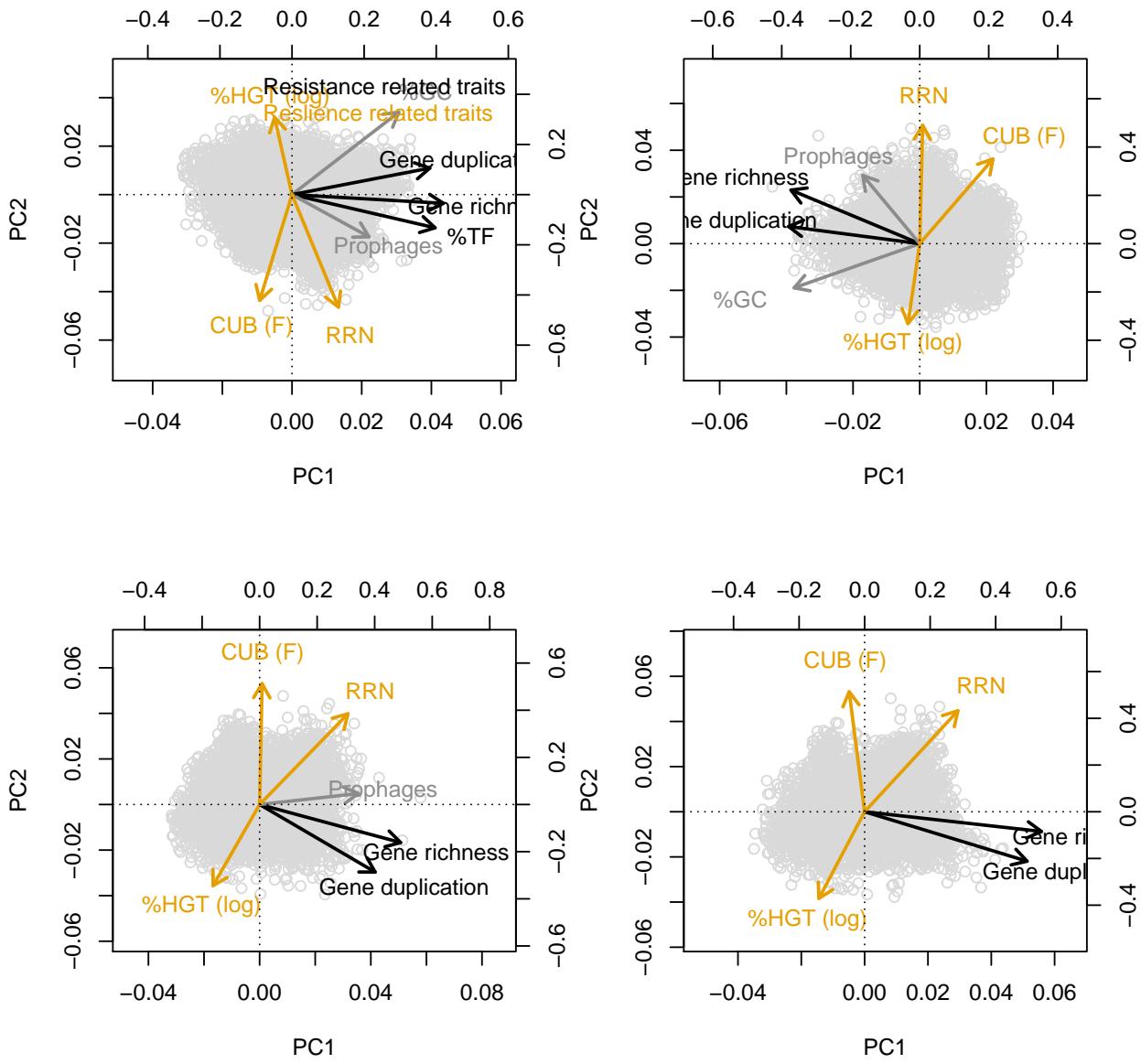
```

4]
plot(pca6.scores, xlim = xlim6, ylim = ylim6, col = "gray85", cex = 1)
par(new = TRUE)
plot(pca6.loadings, axes = FALSE, type = "n", xlim = xlim6 * ratio, ylim = ylim6 *
ratio, xlab = "", ylab = "")
axis(3, col = "black")
axis(4, col = "black")
text(pca6.loadings, labels = rownames(pca6.loadings), cex = 1, col = col)
arrows(0, 0, pca6.loadings[, 1] * 0.8, pca6.loadings[, 2] * 0.8, col = col,
length = 0.1, lwd = 2)
abline(v = 0, h = 0, lty = "dotted")

col = merge(pca5.loadings, color.schema, by.x = 0, by.y = "names.vars.")[,  

4]
plot(pca5.scores, xlim = xlim5, ylim = ylim5, col = "gray85", cex = 1)
par(new = TRUE)
plot(pca5.loadings, axes = FALSE, type = "n", xlim = xlim5 * ratio, ylim = ylim5 *
ratio, xlab = "", ylab = "")
axis(3, col = "black")
axis(4, col = "black")
text(pca5.loadings, labels = rownames(pca5.loadings), cex = 1, col = col)
arrows(0, 0, pca5.loadings[, 1] * 0.8, pca5.loadings[, 2] * 0.8, col = col,
length = 0.1, lwd = 2)
abline(v = 0, h = 0, lty = "dotted")

```



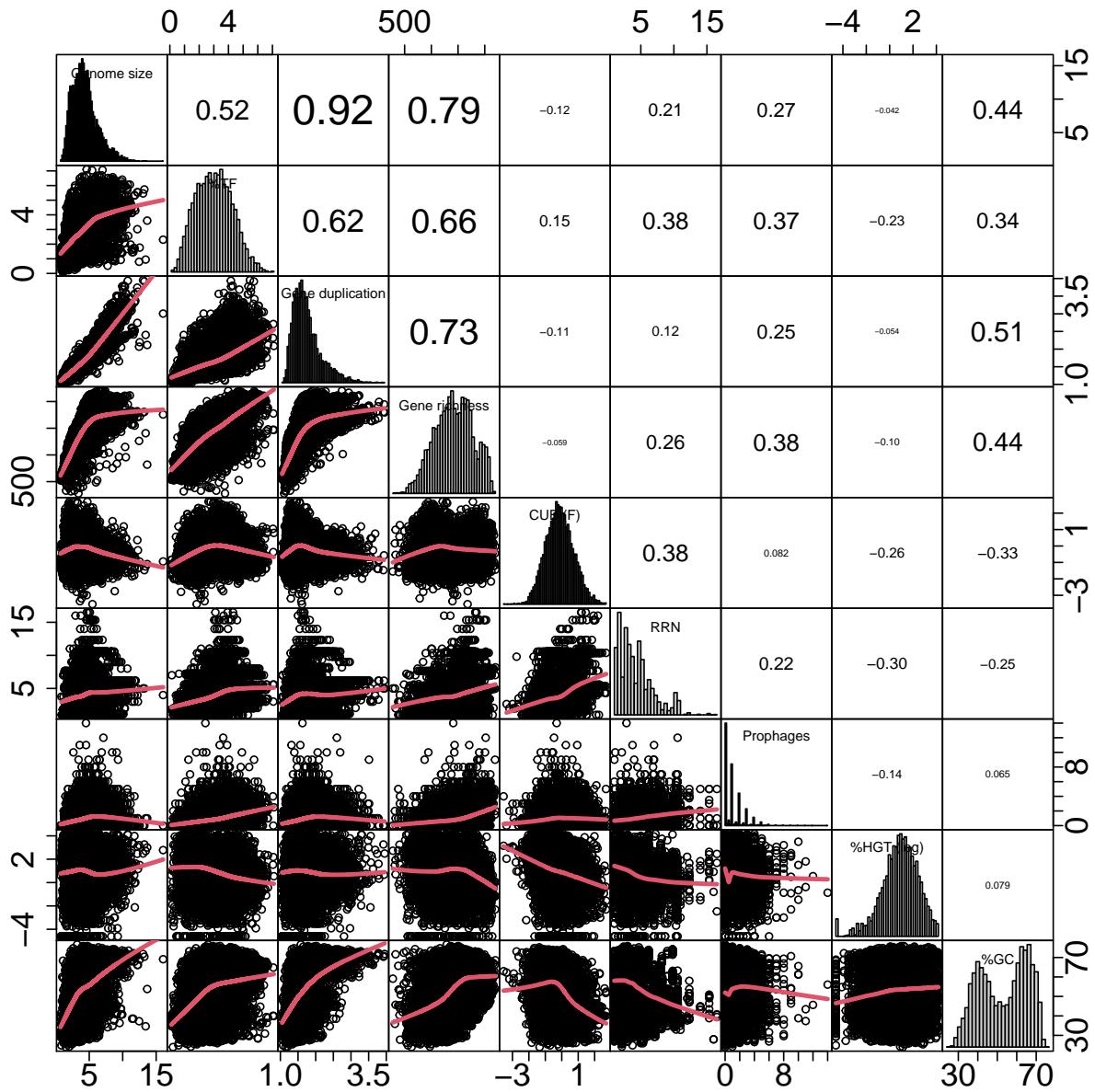
```
## pdf
## 2
```

Figure S4: The random removal of 1,2,3 or 4 variable from the principal component analyses presented in Figure 3 demonstrates that the spatial patterns of resistance versus resilience related genomic traits are robust against the removal of individual variables from the dataset. (The positions of some text labels were later adjusted with an external graphic program)

7 Pairwise correlations (traits averaged at the species level)

7.1 Pairwise scatterplots among 9 genomic traits (Figure 4)

```
# Figure 4
chart.Correlation.mod(vars, histogram = TRUE, pch = 19, method = c("spearman"))
```



```

## [1] "Genome size"      "%TF"           "Gene duplication" "Gene richness"
## [5] "CUB (F)"         "RRN"           "Prophages"       "%HGT (log)"
## [9] "%GC"

## pdf
## 2

```

Figure 4: Pairwise scatterplots among genomic traits. (The positions and colors of some text labels were later adjusted with an external graphic program)

7.2 Pairwise scatterplots of resistance realted tratis versus CUB parameters (Figure 5)

```

# Figure 5
par(mfrow = c(3, 3))

```

```

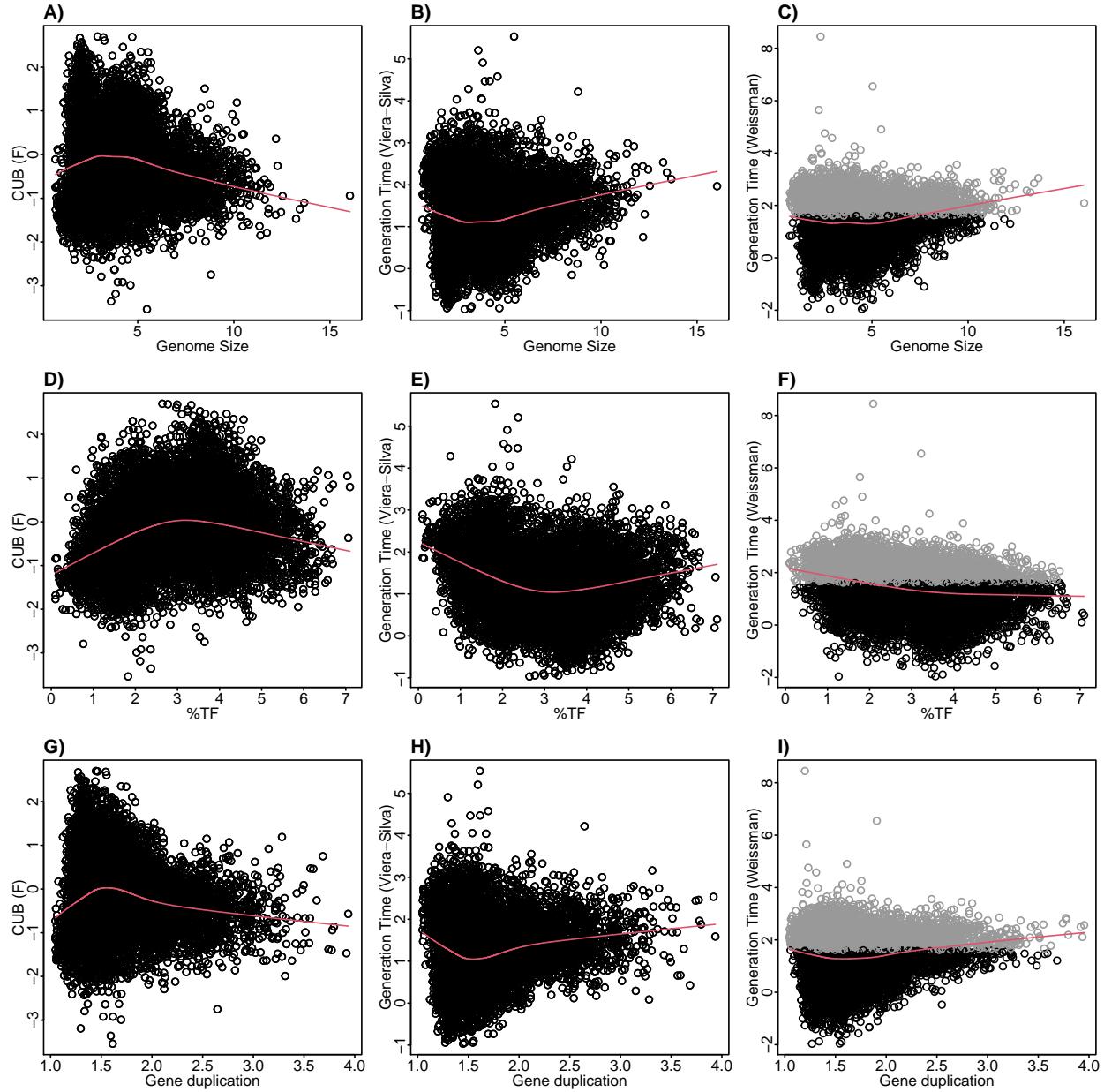
par(mar = c(2.5, 2.5, 1.5, 0.2))
par(mgp = c(0.9, 0.1, 0))
# A)
plot(gtraits.sp$Genome.size, gtraits.sp$CUB.F., ylab = "CUB (F)", xlab = "Genome Size",
      tck = -0.01)
panel.smooth(gtraits.sp$Genome.size, gtraits.sp$CUB.F.)
title("A)", adj = 0, line = 0.4)
# B)
plot(gtraits.sp$Genome.size, gtraits.sp$log.d, ylab = "Generation Time (Viera-Silva)",
      xlab = "Genome Size", tck = -0.01)
panel.smooth(gtraits.sp$Genome.size, gtraits.sp$log.d)
title("B)", adj = 0, line = 0.4)
# C)
plot(gtraits.sp$Genome.size, gtraits.sp$log.d.gRodon, ylab = "Generation Time (Weissman)",
      xlab = "Genome Size", tck = -0.01, col = ifelse(gtraits.sp$log.d.gRodon >=
          log(5), "gray60", "black"))
good <- !(is.na(gtraits.sp$log.d.gRodon) | is.na(gtraits.sp$Genome.size))
lines(lowess(gtraits.sp$log.d.gRodon[good] ~ gtraits.sp$Genome.size[good],
      f = 2/3, iter = 3), col = 2)
title("C)", adj = 0, line = 0.4)
# D)
plot(gtraits.sp$X.TF, gtraits.sp$CUB.F., ylab = "CUB (F)", xlab = "%TF",
      tck = -0.01)
panel.smooth(gtraits.sp$X.TF, gtraits.sp$CUB.F.)
# legend('topright', paste('max: ', round(exTF_F,2), '%'), bty='n')
title("D)", adj = 0, line = 0.4)
# E)
plot(gtraits.sp$X.TF, gtraits.sp$log.d, ylab = "Generation Time (Viera-Silva)",
      xlab = "%TF", tck = -0.01)
panel.smooth(gtraits.sp$X.TF, gtraits.sp$log.d)
title("E)", adj = 0, line = 0.4)
# F)
plot(gtraits.sp$X.TF, gtraits.sp$log.d.gRodon, ylab = "Generation Time (Weissman)",
      xlab = "%TF", tck = -0.01, col = ifelse(gtraits.sp$log.d.gRodon >=
          log(5), "gray60", "black"))
good <- !(is.na(gtraits.sp$log.d.gRodon) | is.na(gtraits.sp$X.TF))
lines(lowess(gtraits.sp$log.d.gRodon[good] ~ gtraits.sp$X.TF[good], f = 2/3,
      iter = 3), col = 2)
title("F)", adj = 0, line = 0.4)
# G)
plot(gtraits.sp$Gene.duplication, gtraits.sp$CUB.F., ylab = "CUB (F)",
      xlab = "Gene duplication", tck = -0.01)
panel.smooth(gtraits.sp$Gene.duplication, gtraits.sp$CUB.F.)
title("G)", adj = 0, line = 0.4)
# H)
plot(gtraits.sp$Gene.duplication, gtraits.sp$log.d, ylab = "Generation Time (Viera-Silva)",
      xlab = "Gene duplication", tck = -0.01)
panel.smooth(gtraits.sp$Gene.duplication, gtraits.sp$log.d)
title("H)", adj = 0, line = 0.4)
# I)
plot(gtraits.sp$Gene.duplication, gtraits.sp$log.d.gRodon, ylab = "Generation Time (Weissman)",
      xlab = "Gene duplication", tck = -0.01, col = ifelse(gtraits.sp$log.d.gRodon >=
          log(5), "gray60", "black"))

```

```

good <- !(is.na(gtraits.sp$log.d.gRodon) | is.na(gtraits.sp$Gene.duplication))
lines(lowess(gtraits.sp$log.d.gRodon[good] ~ gtraits.sp$Gene.duplication[good],
f = 2/3, iter = 3), col = 2)
title("I)", adj = 0, line = 0.4)

```



```

## pdf
## 2

```

Figure 5: Scatterplots displaying pairwise relationships between the codon usage bias (F, Vieira-Silva and Rocha, 2010) the generation time (log-transformed) estimated as detailed in Vieira-Silva et al. (Vieira-Silva and Rocha, 2010) and the generation time (log-transformed) estimated using the R package gRodon (v0.0.0.9000, Weissman et al., 2021) against genomes size, %TF and gene duplication. Genomes with low codon usage bias resulting in predicted generations times (gRodon) exceeding 5 h (highlighted in gray) represent species with generations times >5 h, while the exact value is inaccurate. The absence of genomes with elevated generation

times at the extremes of the value ranges for the displayed resistance related traits is accordingly independent from these inaccuracies.

8 Partial correlations (Table 1)

```

gtraits1 <- gtraits.sp[gtraits.sp$Genome.size<=4,] #species with genome size <=4
gtraits2 <- gtraits.sp[gtraits.sp$Genome.size>=5,] #species with genome size >=5

partial.rho <- data.frame(rho.1=c(cor.test(gtraits1$Genome.size, gtraits1$CUB.F.,
                                              method="spearman")[[4]],
                                     cor.test(gtraits1$Genome.size, gtraits1$RRN_rrnDB,
                                              method="spearman")[[4]],
                                     cor.test(gtraits1$Gene.duplication, gtraits1$CUB.F.,
                                              method="spearman")[[4]],
                                     cor.test(gtraits1$Gene.duplication, gtraits1$RRN_rrnDB,
                                              method="spearman")[[4]],
                                     cor.test(gtraits1$X.TF, gtraits1$CUB.F.,
                                              method="spearman")[[4]],
                                     cor.test(gtraits1$X.TF, gtraits1$RRN_rrnDB,
                                              method="spearman")[[4]]),

rho.2=c(cor.test(gtraits2$Genome.size, gtraits2$CUB.F.,
                  method="spearman")[[4]],
        cor.test(gtraits2$Genome.size, gtraits2$RRN_rrnDB,
                  method="spearman")[[4]],
        cor.test(gtraits2$Gene.duplication, gtraits2$CUB.F.,
                  method="spearman")[[4]],
        cor.test(gtraits2$Gene.duplication, gtraits2$RRN_rrnDB,
                  method="spearman")[[4]],
        cor.test(gtraits2$X.TF, gtraits2$CUB.F.,
                  method="spearman")[[4]],
        cor.test(gtraits2$X.TF, gtraits2$RRN_rrnDB,
                  method="spearman")[[4]]))

rownames(partial.rho) <- c('Genome size vs CUB', 'Genome size vs RRN',
                           'Gene duplication vs CUB', 'Gene duplication vs RRN',
                           '%TF vs CUB', '%TF vs RRN')

partial.p <- data.frame(p.1=c(cor.test(gtraits1$Genome.size, gtraits1$CUB.F.,
                                         method="spearman")[[3]],
                               cor.test(gtraits1$Genome.size, gtraits1$RRN_rrnDB,
                                         method="spearman")[[3]],
                               cor.test(gtraits1$Gene.duplication, gtraits1$CUB.F.,
                                         method="spearman")[[3]],
                               cor.test(gtraits1$Gene.duplication, gtraits1$RRN_rrnDB,
                                         method="spearman")[[3]],
                               cor.test(gtraits1$X.TF, gtraits1$CUB.F.,
                                         method="spearman")[[3]],

```

```

        method="spearman")[[3]],
cor.test(gtraits1$X.TF, gtraits1$RRN_rrnDB,
        method="spearman")[[3]]),

p.2=c(cor.test(gtraits2$Genome.size, gtraits2$CUB.F.,
        method="spearman")[[3]],
cor.test(gtraits2$Genome.size, gtraits2$RRN_rrnDB,
        method="spearman")[[3]],

cor.test(gtraits2$Gene.duplication, gtraits2$CUB.F.,
        method="spearman")[[3]],
cor.test(gtraits2$Gene.duplication, gtraits2$RRN_rrnDB,
        method="spearman")[[3]],

cor.test(gtraits2$X.TF, gtraits2$CUB.F.,
        method="spearman")[[3]],
cor.test(gtraits2$X.TF, gtraits2$RRN_rrnDB,
        method="spearman")[[3]]))

rownames(partial.p) <- c('Genome size vs CUB', 'Genome size vs RRN',
        'Gene duplication vs CUB', 'Gene duplication vs RRN',
        '%TF vs CUB', '%TF vs RRN')
partial.p <- partial.p *12 #correct for multiple pairwise comparisons
cbind(partial.rho, partial.p)

```

	rho.1	rho.2	p.1	p.2
## Genome size vs CUB	0.05498845	-0.27456072	5.366894e-03	1.460472e-44
## Genome size vs RRN	0.08957200	-0.02276168	2.658807e-05	3.655556e+00
## Gene duplication vs CUB	0.12251446	-0.22906158	5.191756e-14	7.633830e-31
## Gene duplication vs RRN	0.07371961	-0.15904954	1.673868e-03	8.422135e-12
## %TF vs CUB	0.44281760	-0.12183041	4.261743e-194	7.043748e-09
## %TF vs RRN	0.38248210	0.22648473	1.670563e-92	9.207386e-24

Table 1 displays correlation coefficients and p-values between selected resistance and resilience related traits for species with small (≤ 4 Mbp) and large (≥ 5 Mbp) genomes

9 Habitat specific PCAs (traits considered at the genome level, Figure 6)

To visualize trait covariation patterns in dependence of the habitat type we sorted genomes based on information given in the column ‘Habitat’ of Table 1 into three habitat types: all genomes containing the strings ‘soil’ or ‘rhizosphere’ in the habitat description where classified as originating from *soil habitats*; all genomes containing the strings ‘aquatic’ or ‘marine’ or ‘water’ where classified as originating from *aquatic habitats*; all genomes containing the strings ‘oral’ or ‘stomach’ or ‘gut’ or ‘intestinal’ or ‘feces’ were classified as originating from the intestinal tract. Genomes that could not assigned to either of these three habitats were not further classified. Trait values were not summarized at the species level because habitat information was given at the genome level.

```

vars.habitat <- gtraits %>%
  mutate(log.HGT_per= log(per.HGT.corr+0.01)) %>% #log(x+0.01) data transformation
  mutate(Habitat.color = ifelse (grepl('aquatic|marine|water', gtraits$Habitat, ignore.case=T),
        'steelblue3','gray85')) %>%
  mutate(Habitat.color = ifelse (grepl('soil|rhizosphere', gtraits$Habitat, ignore.case=T),
        'tomato4',Habitat.color)) %>%

```

```

mutate(Habitat.color = ifelse (grepl('intestinal|feces|stomach|gut|oral', gtraits$Habitat,
                                      ignore.case=T), 'lightpink3',Habitat.color)) %>%
arrange((Habitat.color)) %>%
select (Genome.size,X.TF, Gene.duplication, Gene.richness,CUB.F.,RRN_rrnDB,Prophages.corr,
        log.HGT_per,X.GC, Habitat.color)
colnames(vars.habitat) <- c('Genome size', '%TF', 'Gene duplication', 'Gene richness',
                           'CUB (F)', 'RRN', 'Prophages', '%HGT (log)', '%GC', 'Habitat.color')
vars.habitat

## # A tibble: 17,856 x 10
##   `Genome size` `%TF` `Gene duplication` `Gene richness` `CUB (F)`    RRN
##       <dbl> <dbl>          <dbl>           <int>      <dbl> <dbl>
## 1        2.09  1.60            1.40         745     -0.9    NA
## 2        1.46  1.60            1.27         790     -1.32   NA
## 3        3.26  1.34            1.82         925     -1.2    NA
## 4        1.86  1.93            1.36        1084    -1.19    1
## 5        1.61  1.48            1.29         916     -1.21   NA
## 6        1.42  1.47            1.26         947     -1.32   NA
## 7        4.67  2.32            2.04        1256     0.27    NA
## 8        5.19  2.57            2.06        1239     0.17    NA
## 9        3.70  NA              NA             NA     NA     1
## 10       3.64  2.30            1.76        1199     0.54    1
## # ... with 17,846 more rows, and 4 more variables: Prophages <int>,
## #   `%HGT (log)` <dbl>, `%GC` <dbl>, Habitat.color <chr>

vars.aqua <- vars.habitat[vars.habitat$Habitat.color=='steelblue3',]
vars.soil <- vars.habitat[vars.habitat$Habitat.color=='tomato4',]
vars.digest <- vars.habitat[vars.habitat$Habitat.color=='lightpink3',]

# Mean genome size in each habitat type
mean.gs <- vars.habitat[,c(1,10)] %>%
tibble() %>%
mutate(Habitat = ifelse (Habitat.color=='steelblue3','aquatic','undefined')) %>%
mutate(Habitat = ifelse (Habitat.color=='tomato4','soil',Habitat)) %>%
mutate(Habitat = ifelse (Habitat.color=='lightpink3','digestive tract',Habitat)) %>%
select(-Habitat.color) %>%
group_by(Habitat) %>%
summarise_each(funs(mean, sd))
mean.gs

## # A tibble: 4 x 3
##   Habitat      mean     sd
##   <chr>      <dbl>  <dbl>
## 1 aquatic     3.90  1.51
## 2 digestive tract 3.35  1.46
## 3 soil        5.73  1.94
## 4 undefined    4.15  1.76

# Principal component analyses
pca<-prcomp(na.omit(vars.habitat[,-10]),center=TRUE, scale=TRUE)
pca.aqua<-prcomp(na.omit(vars.aqua[,-10]),center=TRUE, scale=TRUE)
pca.soil<-prcomp(na.omit(vars.soil[,-10]),center=TRUE, scale=TRUE)
pca.digest<-prcomp(na.omit(vars.digest[,-10]),center=TRUE, scale=TRUE)

# Proportion variance explained in components 1 and 2

```

```

prop.1<-summary(pca)$importance[2]*100
prop.2<-summary(pca)$importance[5]*100
prop.aqua.1<-summary(pca.aqua)$importance[2]*100
prop.aqua.2<-summary(pca.aqua)$importance[5]*100
prop.soil.1<-summary(pca.soil)$importance[2]*100
prop.soil.2<-summary(pca.soil)$importance[5]*100
prop.digest.1<-summary(pca.digest)$importance[2]*100
prop.digest.2<-summary(pca.digest)$importance[5]*100

# Extract the eigenvectors and the loading factors and normalize PC axes to unit length
pca.scores<-pca$x[,1:2]
pca.loadings<-pca$rotation[,1:2]
pca.scores[,1]<-pca.scores[,1]*(1/sqrt(sum(pca.scores[,1]^2)))
pca.scores[,2]<-pca.scores[,2]*(1/sqrt(sum(pca.scores[,2]^2)))

pca.aqua.scores<-pca.aqua$x[,1:2]
pca.aqua.loadings<-pca.aqua$rotation[,1:2]
pca.aqua.scores[,1]<-pca.aqua.scores[,1]*(1/sqrt(sum(pca.aqua.scores[,1]^2)))
pca.aqua.scores[,2]<-pca.aqua.scores[,2]*(1/sqrt(sum(pca.aqua.scores[,2]^2)))

pca.soil.scores<-pca.soil$x[,1:2]
pca.soil.loadings<-pca.soil$rotation[,1:2]
pca.soil.scores[,1]<-pca.soil.scores[,1]*(1/sqrt(sum(pca.soil.scores[,1]^2)))
pca.soil.scores[,2]<-pca.soil.scores[,2]*(1/sqrt(sum(pca.soil.scores[,2]^2)))

pca.digest.scores<-pca.digest$x[,1:2]
pca.digest.loadings<-pca.digest$rotation[,1:2]
pca.digest.scores[,1]<-pca.digest.scores[,1]*(1/sqrt(sum(pca.digest.scores[,1]^2)))
pca.digest.scores[,2]<-pca.digest.scores[,2]*(1/sqrt(sum(pca.digest.scores[,2]^2)))

# Set scaling parameters for biplots
sc<-1.5
xlim <- unsigned.range(pca.scores[,1])*sc
ylim <- unsigned.range(pca.scores[,2])*sc
ratio <-max(abs(unsigned.range(pca.loadings[,1])[1]
                 -unsigned.range(pca.loadings[,1])[2])/abs(xlim[1]-xlim[2]),
            abs(unsigned.range(pca.loadings[,2])[1]
                 -unsigned.range(pca.loadings[,2])[2])/abs(ylim[1]-ylim[2]))*sc/1.3

xlim.aqua <- unsigned.range(pca.aqua.scores[,1])*sc
ylim.aqua <- unsigned.range(pca.aqua.scores[,2])*sc
ratio.aqua <-max(abs(unsigned.range(pca.aqua.loadings[,1])[1]
                     -unsigned.range(pca.aqua.loadings[,1])[2])/abs(xlim.aqua[1]
                                                               -xlim.aqua[2]),
                  abs(unsigned.range(pca.aqua.loadings[,2])[1]
                     -unsigned.range(pca.aqua.loadings[,2])[2])/abs(ylim.aqua[1]
                                                               -ylim.aqua[2]))*sc/1.3

xlim.soil <- unsigned.range(pca.soil.scores[,1])*sc
ylim.soil <- unsigned.range(pca.soil.scores[,2])*sc
ratio.soil <-max(abs(unsigned.range(pca.soil.loadings[,1])[1]
                     -unsigned.range(pca.soil.loadings[,1])[2])/abs(xlim.soil[1]
                                                               -xlim.soil[2]),
                  abs(unsigned.range(pca.soil.loadings[,2])[1]
                     -unsigned.range(pca.soil.loadings[,2])[2])/abs(ylim.soil[1]
                                                               -ylim.soil[2]))*sc/1.3

```

```

abs(unsigned.range(pca.soil.loadings[,2])[1]
    -unsigned.range(pca.soil.loadings[,2])[2])/abs(ylim.soil[1]
        -ylim.soil[2]))*sc/1.3

xlim.digest <- unsigned.range(pca.digest.scores[,1])*sc
ylim.digest <- unsigned.range(pca.digest.scores[,2])*sc
ratio.digest <-max(abs(unsigned.range(pca.digest.loadings[,1])[1]
    -unsigned.range(pca.digest.loadings[,1])[2])/abs(xlim.digest[1]
        -xlim.digest[2])),
abs(unsigned.range(pca.digest.loadings[,2])[1]
    -unsigned.range(pca.digest.loadings[,2])[2])/abs(ylim.digest[1]
        -ylim.digest[2]))*sc/1.3

# Figure 6
par(mfrow=c(2,2))
par(mgp=c(1.5,0.4,0))

col=c(rep("black",4), rep("#E69F00",2), c("gray55","#E69F00","gray55"))
col.genomes <- na.omit(vars.habitat) %>% pull(Habitat.color)
plot(pca.scores,xlim=xlim,ylim=ylim,col=col.genomes,cex=1,
      ylab=paste("PC2 (",round(prop.2,1)," % contribution to total variability"),
      xlab=paste("PC1 (",round(prop.1,1)," % contribution to total variability")))
axis(1, col = "gray85")
axis(2, col = "gray85")
title("A)", line=2.5, adj=0)
legend('topleft',c("Aquatic","Soil","Digestive tract","NA"),pch = c(1,1,1,1),
       col=c('steelblue3','tomato4','lightpink3',"gray85"),bty="n")
legend('topright',c("Resistance related traits","Resilience related traits"),
       text.col=c('black','#E69F00'),bty="n")
legend('bottomleft', paste(c('Mean genome size: '),
                           round(mean(vars.habitat[,1], drop=T)),1),
       c('Mpb')),bty="n")
par(new=TRUE)
plot(pca.loadings,axes=FALSE,type="n",xlim=xlim*ratio,ylim=ylim*ratio,
      xlab = "", ylab = "")
axis(3, col = "black")
axis(4, col = "black")
text(pca.loadings, labels = rownames(pca.loadings), cex = 1, col = col)
arrows(0, 0, pca.loadings[, 1] * 0.8, pca.loadings[, 2] * 0.8, col = col,
       length = 0.1,lwd =2)
abline(v=0,h=0,lty="dotted")

plot(pca.soil.scores,xlim=xlim.soil,ylim=ylim.soil,col='tomato4',cex=1,
      ylab=paste("PC2 (",round(prop.soil.2,1)," % contribution to total variability"),
      xlab=paste("PC1 (",round(prop.soil.1,1)," % contribution to total variability")))
axis(1, col = "gray85")
axis(2, col = "gray85")
title("B)", line=2.5, adj=0)
legend('bottomleft', paste(c('Mean genome size: '),
                           round(mean.gs[3,2,drop=T],1),
                           c('Mbp')),bty="n")
par(new=TRUE)
plot(pca.soil.loadings,axes=FALSE,type="n",xlim=xlim.soil*ratio.soil,
      ylim=ylim.soil*ratio.soil,xlab = "", ylab = "")

```

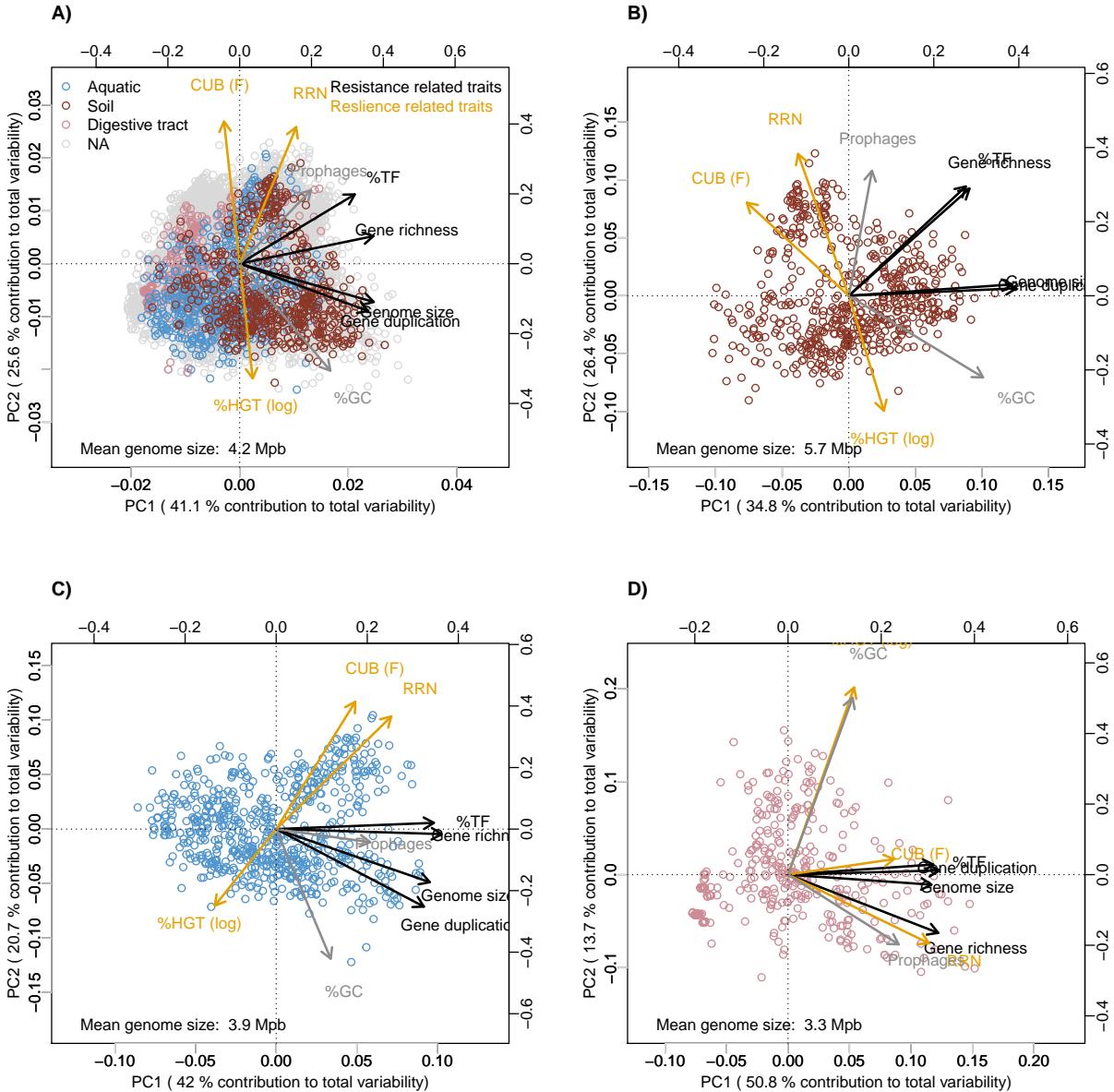
```

axis(3, col = "black")
axis(4, col = "black")
text(pca.soil.loadings, labels = rownames(pca.soil.loadings), cex = 1, col = col)
arrows(0, 0, pca.soil.loadings[, 1] * 0.8, pca.soil.loadings[, 2] * 0.8, col = col,
       length = 0.1,lwd =2)
abline(v=0,h=0,lty="dotted")

plot(pca.aqua.scores,xlim=xlim.aqua,ylim=ylim.aqua,col='steelblue3',cex=1,
      ylab=paste("PC2 (",round(prop.aqua.2,1),"% contribution to total variability)"),
      xlab=paste("PC1 (",round(prop.aqua.1,1),"% contribution to total variability)"))
axis(1, col = "gray85")
axis(2, col = "gray85")
title("C)", line=2.5, adj=0)
legend('bottomleft', paste(c('Mean genome size: '),round(mean.gs[1,2,drop=T],1),
                           c('Mpb')),bty="n")
par(new=TRUE)
plot(pca.aqua.loadings,axes=FALSE,type="n",xlim=xlim.aqua*ratio.aqua,
      ylim=ylim.aqua*ratio.aqua,
      xlab = "", ylab = "")
axis(3, col = "black")
axis(4, col = "black")
text(pca.aqua.loadings, labels = rownames(pca.aqua.loadings), cex = 1, col = col)
arrows(0, 0, pca.aqua.loadings[, 1] * 0.8, pca.aqua.loadings[, 2] * 0.8, col = col,
       length = 0.1,lwd =2)
abline(v=0,h=0,lty="dotted")

plot(pca.digest.scores,xlim=xlim.digest,ylim=ylim.digest,col='lightpink3',cex=1,
      ylab=paste("PC2 (",round(prop.digest.2,1),"% contribution to total variability)"),
      xlab=paste("PC1 (",round(prop.digest.1,1),"% contribution to total variability)"))
axis(1, col = "gray85")
axis(2, col = "gray85")
title("D)", line=2.5, adj=0)
legend('bottomleft', paste(c('Mean genome size: '), round(mean.gs[2,2,drop=T],1),
                           c('Mpb')),bty="n")
par(new=TRUE)
plot(pca.digest.loadings,axes=FALSE,type="n",xlim=xlim.digest*ratio.digest,
      ylim=ylim.digest*ratio.digest,
      xlab = "", ylab = "")
axis(3, col = "black")
axis(4, col = "black")
text(pca.digest.loadings, labels = rownames(pca.digest.loadings), cex = 1, col = col)
arrows(0, 0, pca.digest.loadings[, 1] * 0.8, pca.digest.loadings[, 2] * 0.8, col = col,length = 0.1,
       lwd =2)
abline(v=0,h=0,lty="dotted")

```



```
## pdf
## 2
```

Figure 6: Principal component analyses illustrating covariations among genomic traits from JGI/IMG prokaryotic genomes dependence on the habitat type. (The positions of some text labels were later adjusted with an external graphic program)

10 Mantel correlograms (Figure 7)

Input files for Figure 7 were created as detailed in scripts_phylogenetic.signal.md

```
# Change column names for Figure labels
mpm.2$trait <- c("Genome size")
tibble(mpm.2) #inspect input file
```

```

## # A tibble: 8 x 6
##   clases          rM pvalues pval.Bonferroni    pd trait
##   <chr>        <dbl>  <dbl>            <dbl> <dbl> <chr>
## 1 0     - 0.25    0.0953  0.00498      0.00498  0.25 Genome size
## 2 0.25 - 0.5    0.0570  0.00498      0.00995  0.5  Genome size
## 3 0.5  - 0.75   0.0535  0.00498      0.0149   0.75 Genome size
## 4 0.75 - 1      0.0397  0.00498      0.0199   1    Genome size
## 5 1     - 1.5    0.0342  0.00498      0.0249   1.5  Genome size
## 6 1.5  - 2      -0.000217 0.473     2.84     2    Genome size
## 7 2     - 2.5    0.00419  0.0149    0.104    2.5  Genome size
## 8 2.5  - 3      -0.0422  0.00498      0.0398   3    Genome size

mpm.3$trait <- c("%GC")
mpm.4$trait <- c("RRN")
mpm.5$trait <- c("%HGT (log)")
mpm.6$trait <- c("CUB (F)")
mpm.7$trait <- c("Generation time (gRodon, log)")
mpm.8$trait <- c("Gene duplication")
mpm.9$trait <- c("Gene richness")
mpm.10$trait <- c("%TF")
mpm.11$trait <- c("Prophages")

# Set common ylim range for all plots
ylim = range(rbind(mpm.2, mpm.3, mpm.4, mpm.5, mpm.6, mpm.7, mpm.8, mpm.9,
                    mpm.10, mpm.11[, 2]))

# Figure 7
par(mfrow = c(5, 2))
par(lwd = 1)
par(mar = c(2, 1.8, 2, 0.2))
par(mgp = c(1, 0.2, 0))

mpm <- mpm.2
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &
                                                 mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "rho", xlab = "")
abline(h = 0, lty = 2)

mpm <- mpm.10
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &
                                                 mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "", xlab = "")
abline(h = 0, lty = 2)

mpm <- mpm.8
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &
                                                 mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "rho", xlab = "")
abline(h = 0, lty = 2)

mpm <- mpm.9
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &
                                                 mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "", xlab = "")
abline(h = 0, lty = 2)

mpm <- mpm.6
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &

```

```

mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "rho", xlab = "")
abline(h = 0, lty = 2)

mpm <- mpm.7
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &
  mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "", xlab = "")
abline(h = 0, lty = 2)

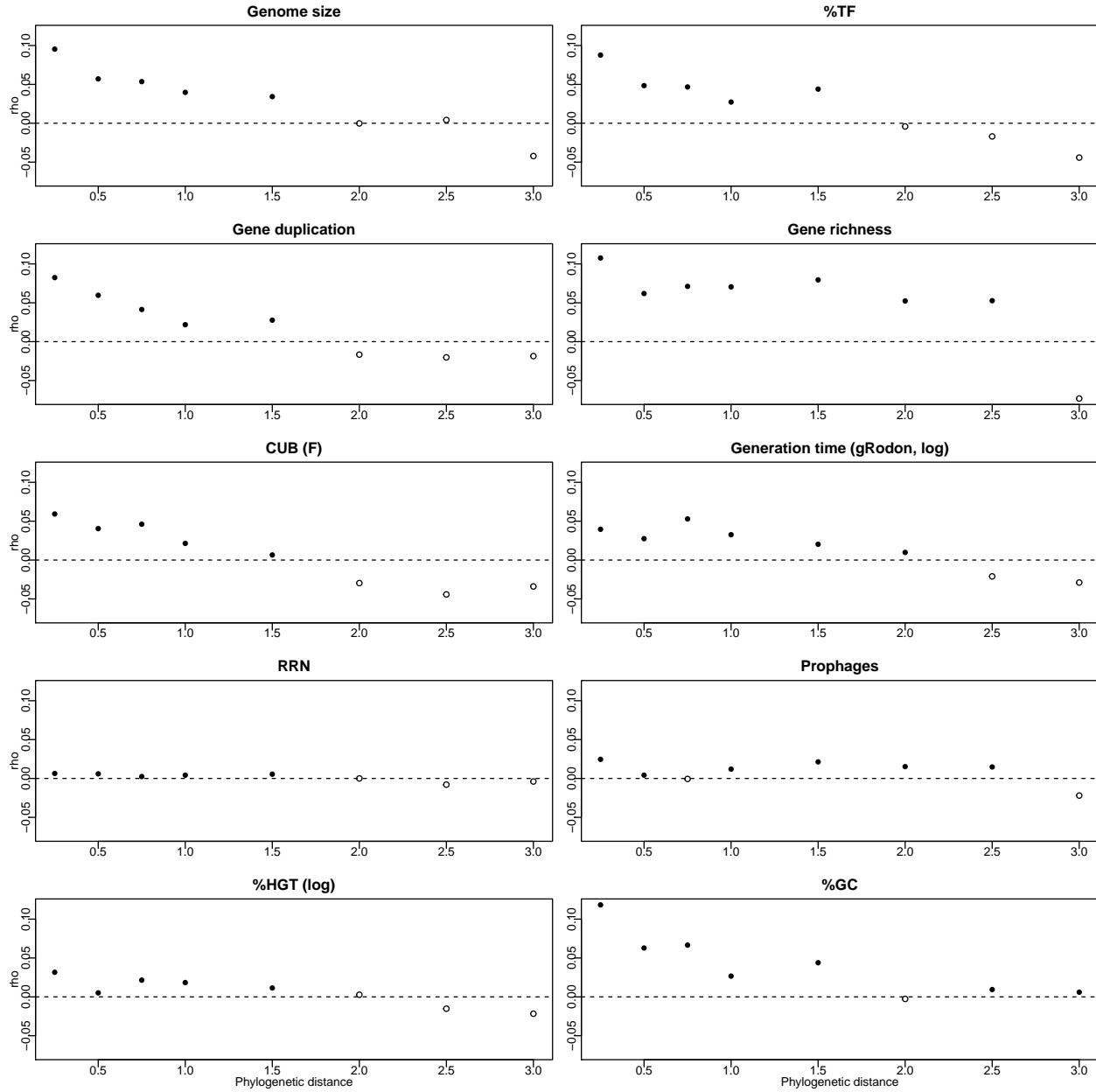
mpm <- mpm.4
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &
  mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "rho", xlab = "")
abline(h = 0, lty = 2)

mpm <- mpm.11
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &
  mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "", xlab = "")
abline(h = 0, lty = 2)

mpm <- mpm.5
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &
  mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "rho", xlab = "Phylogenetic distance")
abline(h = 0, lty = 2)

mpm <- mpm.3
plot(mpm$pd, mpm$rM, ylim = ylim, pch = ifelse(mpm$pval.Bonferroni < 0.1 &
  mpm$rM > 0, 16, 1), main = mpm$trait[1], ylab = "", xlab = "Phylogenetic distance")
abline(h = 0, lty = 2)

```



```
## pdf
## 2
```

Figure 7: Results from Mantel correlograms. Filled data points indicate positive significant correlations (Bonferroni adjusted p -values $Padj < 0.1$) of pairwise distances of traits values against pairwise phylogenetic distances among the reference genomes. The Mantel correlograms were computed for 10,000 randomly selected genomes and with 200 permutations.