Figure 2 and supplements

NSMR

```
library(readr)
library(ggplot2)
library(gridExtra)
library(grid)
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
```

```
##
```

```
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```

```
##
## Attaching package: 'gdata'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

```
library(ggpubr)
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggpubr':
##
##     get_legend
```

```
library(RColorBrewer)
library(pheatmap)
library(tidyr)

#function taken from https://stackoverflow.com/a/14674703
symlog_trans <- function(base = 10, thr = 1, scale = 1){
  trans <- function(x)
    ifelse(abs(x) < thr, x, sign(x) *
```

```
            (thr + scale * suppressWarnings(log(sign(x) * x / thr, base))))

  inv <- function(x)
    ifelse(abs(x) < thr, x, sign(x) *
             base^((sign(x) * x - thr) / scale) * thr)

  breaks <- function(x){
    sgn <- sign(x[which.max(abs(x))])
    if(all(abs(x) < thr))
      pretty_breaks()(x)
    else if(prod(x) >= 0){
      if(min(abs(x)) < thr)
        sgn * unique(c(pretty_breaks()(c(min(abs(x)), thr)),
                       log_breaks(base)(c(max(abs(x)), thr))))
      else
        sgn * log_breaks(base)(sgn * x)
    } else {
      if(min(abs(x)) < thr)
        unique(c(sgn * log_breaks()(c(max(abs(x)), thr)),
                 pretty_breaks()(c(sgn * thr, x[which.min(abs(x))]))))
      else
        unique(c(-log_breaks(base)(c(thr, -x[1])),
                 pretty_breaks()(c(-thr, thr)),
                 log_breaks(base)(c(thr, x[2]))))
    }
  }
  scales::trans_new(paste("symlog", thr, base, scale, sep = "-"), trans, inv, breaks)
}
```

```
df <- readr::read_csv('key_nodes.tidydf.csv')
```

```
## Parsed with column specification:
## cols(
##   node = col_character(),
##   taxon = col_character(),
##   species = col_character(),
##   random = col_character(),
##   block_id = col_double(),
##   iteration = col_double(),
##   density = col_double(),
##   acc_ls = col_character(),
##   all_acc_ls = col_character(),
##   total_density = col_double(),
##   total_genome_length = col_double(),
##   density_ratio = col_double(),
##   multi_sp = col_double(),
##   para = col_character(),
##   mean_dist_pair = col_double(),
##   mean_dist_pair_norm = col_double(),
##   median_dist_pair = col_double(),
##   median_dist_pair_norm = col_double()
## )
```

Change data into factor, to reorganize the names in a manner consistent with between figures

```
df$taxon <- factor(df$taxon, levels=c('Poriferan','Ctenophore','Placozoan','Cnidarian', 'Acoel','Ecdyso
df$node <- as.factor(df$node)
df$random <- as.factor(df$random)
df$species <- factor(df$species, levels = c('CAPOW', 'SALRO', 'AMPQU', 'SYCCI', 'MNELE', 'PLEBA', 'TRIA
df$log10_density_ratio <- log10(df$density_ratio)

df_para_Met <- df %>% dplyr::filter(para == 'para' & node == 'Metazoa')
df_para_Par <- df %>% dplyr::filter(para == 'para' & node == 'Parahoxozoa')
df_para_Pla <- df %>% dplyr::filter(para == 'para' & node == 'Planulozoa')
df_para_Bil <- df %>% dplyr::filter(para == 'para' & node == 'Bilateria')
df_para_Ver <- df %>% dplyr::filter(para == 'para' & node == 'Vertebrata')
df_para_Lop <- df %>% dplyr::filter(para == 'para'  & node == 'Lophotrochozoa')
df_not_para_Met <- df %>% dplyr::filter(para == 'not_para' & node == 'Metazoa')
df_not_para_Par <- df %>% dplyr::filter(para == 'not_para' & node == 'Parahoxozoa')
df_not_para_Pla <- df %>% dplyr::filter(para == 'not_para' & node == 'Planulozoa')
df_not_para_Bil <- df %>% dplyr::filter(para == 'not_para' & node == 'Bilateria')
df_not_para_Ver <- df %>% dplyr::filter(para == 'not_para' & node == 'Vertebrata')
df_not_para_Lop <- df %>% dplyr::filter(para == 'not_para' & node == 'Lophotrochozoa')
```

Here we define a function to make boxplots of the supp figure (by taxon).

```
map_signif_level <- c(`****` = 1e-04, `***` = 0.001,  `**` = 0.01, `*` = 0.05, ns = 1)

make_plot <- function(tbl,
                      key = "observed",
                      comparisons = list(c("observed", "random")),
                      bracket_y = NULL,
                      ylims = c(-2.5 , 2.5)) {

  if(is.null(bracket_y)) {
    h = ylims[2] - ylims[1]
    bracket_y = c(.9,.825,.75)*h + ylims[1]
  }

  size.summary <- tbl %>% dplyr::filter(random == "observed") %>%  dplyr::group_by(taxon) %>%  dplyr::su
  ggplot(tbl, aes_string(x = 'random', y = 'log10_density_ratio', fill = 'random')) +
    geom_boxplot(outlier.shape = NA) +
    facet_grid(~ taxon) +
    theme_cowplot() +
    theme(axis.title.x = element_blank(), axis.text.x = element_blank()) +
    geom_signif(comparisons = comparisons,
                test = "wilcox.test", test.args = list(paired = FALSE, exact = FALSE), na.rm = TRUE,
                map_signif_level = map_signif_level,
                color="black", tip_length = 0.01, size = .5, textsize = 2,
                y_position = bracket_y,  data = NULL) +
    scale_y_continuous(name = "log10(Density ratio)", limits = c(-2.5, 2.5)) +
    theme(legend.title = element_blank(),
          plot.margin = unit(c(1,0,0,0), units='cm'),
          legend.position = 'bottom',
          legend.justification = 'center',
          strip.text = element_text(size = 6, angle = 90, margin = margin(5,0,5,0,'pt')),
          axis.ticks.x = element_blank(),
          axis.title.y = element_text(size = 7),
          axis.text = element_text(size = 6)) +
```

```
    geom_text(data=size.summary, aes(x=1,y=2.2,hjust = 0.5,label = label), size = 2, inherit.aes=F)
}
```

Every make_plot call for all the possibilities. Done so so that we can have ggpubr tests with facetting.

```
p1 <- make_plot(df_not_para_Met)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p2 <- make_plot(df_para_Met)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p3 <- make_plot(df_not_para_Par)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p4 <- make_plot(df_para_Par)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p5 <- make_plot(df_not_para_Pla)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p6 <- make_plot(df_para_Pla)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p7 <- make_plot(df_not_para_Bil)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p8 <- make_plot(df_para_Bil)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p9 <- make_plot(df_not_para_Ver)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p10 <- make_plot(df_para_Ver)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

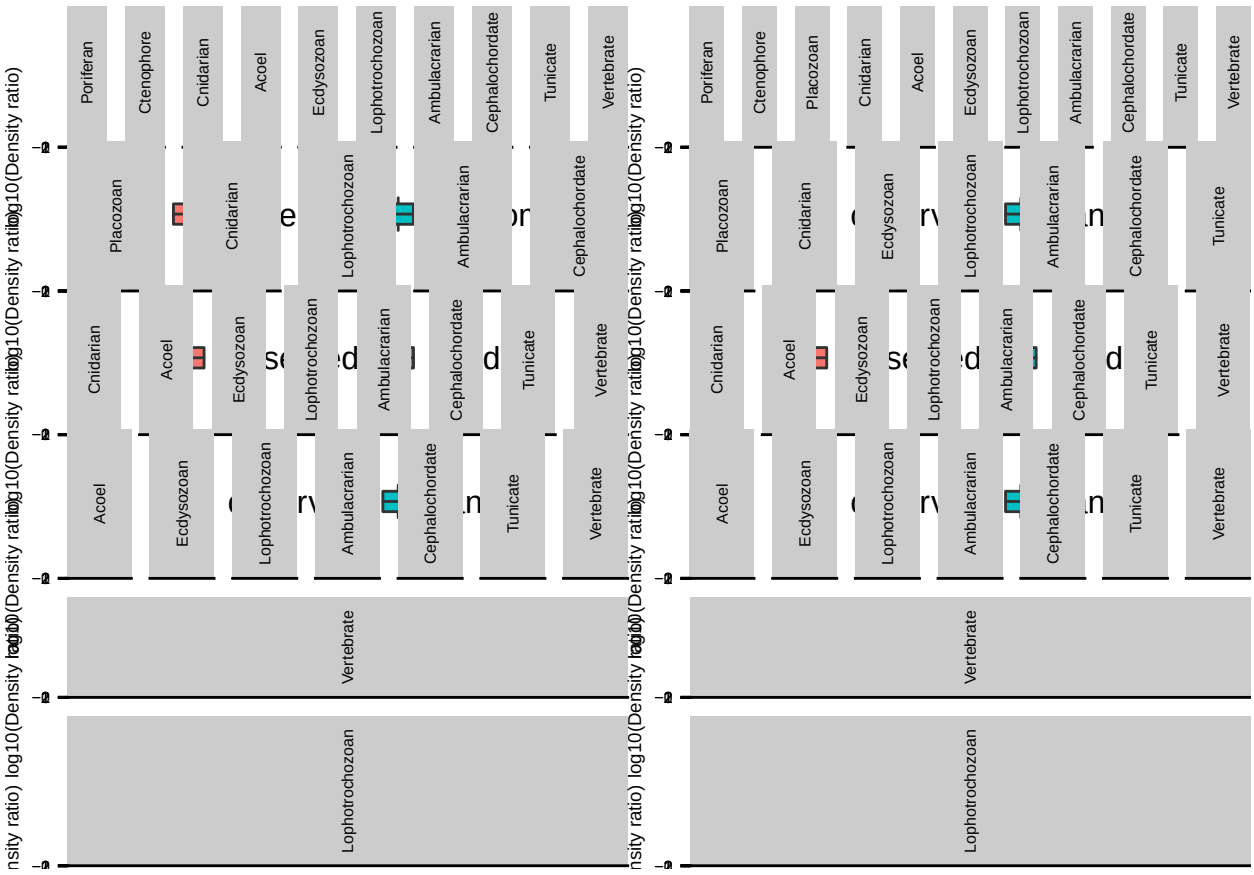
```
p11 <- make_plot(df_not_para_Lop)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
p12 <- make_plot(df_para_Lop)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
gridplot <- gridExtra::grid.arrange(grobs = list(p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12), ncol = 2)
```




```
ggsave(plot = gridplot,
       filename = 'SF3.pdf',
       unit = 'cm',
       width = 30,
       height = 80)
```

Now we do the scatterplots for SF4

```
df <- readr::read_csv('raw_data_scatter.csv')
```

```
## Parsed with column specification:
## cols(
##   multi_sp = col_double(),
##   node = col_character(),
##   Vertebrate = col_double(),
##   Tunicate = col_double(),
##   Cephalochordate = col_double(),
##   Ambulacrarian = col_double(),
##   Lophotrochozoan = col_double(),
##   Ecdysozoan = col_double(),
##   Acoel = col_double(),
##   Cnidarian = col_double(),
##   Placozoan = col_double(),
##   Ctenophore = col_double(),
##   Poriferan = col_double()
```

```
## )
```

```
nodelist <-c('Bilateria', 'Planulozoa', 'Metazoa')
df2 <- df %>% dplyr::filter(node %in% nodelist) %>%
    purrr::discard(~sum(is.na(.x))/length(.x) > 9)
taxons_vars <- colnames(df2)[c(-1,-2)]

# build tables with pairs of values
controlTable <- data.frame(expand.grid(taxons_vars, taxons_vars,
                                       stringsAsFactors = FALSE))
# rename the columns with our taxon names
colnames(controlTable) <- c("x", "y")

# add the key column
controlTable <- cbind(
  data.frame(pair_key = paste(controlTable[[1]], controlTable[[2]]),
             stringsAsFactors = FALSE),
  controlTable)

#Now I can create the new data frame, using the cdata funciton rowrecs_to_blocks(). I'll also carry alo

df2_aug = cdata::rowrecs_to_blocks(
  df2,
  controlTable,
  columnsToCopy = "node")

#let's remove taxon pairs where there is NAs
df2_aug <- na.omit(df2_aug)

splt <- strsplit(df2_aug$pair_key, split = " ", fixed = TRUE)
df2_aug$xv <- vapply(splt, function(si) si[[1]], character(1))
df2_aug$yv <- vapply(splt, function(si) si[[2]], character(1))


# reorder the key columns to be the same order
# as the taxons_vars
df2_aug$xv <- factor(as.character(df2_aug$xv),
                           taxons_vars)
df2_aug$yv <- factor(as.character(df2_aug$yv),
                           taxons_vars)
df2_aug <- df2_aug %>% dplyr::filter(xv != yv)
```

now for the big scatterplot

```
p <- ggplot(df2_aug, aes(x = x, y = y)) +
  geom_point(aes(color = node, shape = node)) +
  facet_grid(yv~xv, scale = "free") +
  scale_y_continuous(trans = symlog_trans(), breaks = c(-10,-1,0,1,10), limits = c(-10.0, 10.0)) +
  scale_x_continuous(trans = symlog_trans(), breaks = c(-10,-1,0,1,10), limits = c(-10.0, 10.0)) +
  ylab(NULL) +
  xlab(NULL) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  ggthemes::theme_gdocs() +
  ggpubr::stat_cor(method = 'spearman', label.x = -8, label.y = 8)
```

```
ggsave('SF4_scatter_density_deviation.pdf',
       plot = p,
       unit = 'cm',
       width = 40,
       height = 40)
```

```
## Warning: Removed 16 rows containing non-finite values (stat_cor).
```

```
## Warning: Removed 16 rows containing missing values (geom_point).
```

```
## Warning: Removed 86 rows containing missing values (geom_text).
```

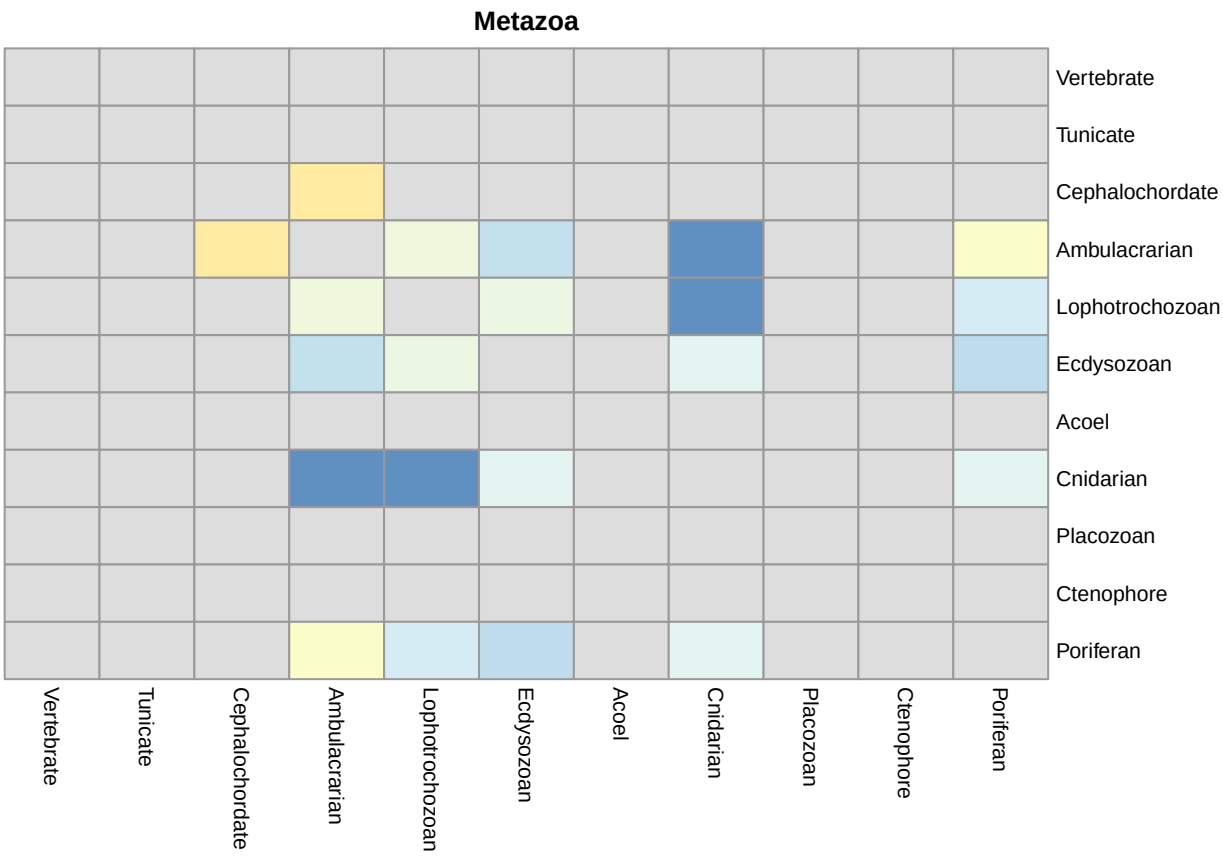
Correlation of the density deviation (ie. relative change of block density ratio)

```
df_Bila <- df %>% dplyr::filter(node == 'Bilateria')
df_Planu <- df %>% dplyr::filter(node == 'Planulozoa')
df_Meta <- df %>% dplyr::filter(node == 'Metazoa')

bila <- taxons_vars[c(1,2,3,4,5,6,7)]
planu <- taxons_vars[c(1,2,3,4,5,6,7,8)]
meta <- taxons_vars

corr_matrix_Bila <- matrix(nrow = length(bila), ncol = length(bila))
colnames(corr_matrix_Bila) <- bila
rownames(corr_matrix_Bila) <- bila
pval_corr_matrix_Bila <- corr_matrix_Bila

corr_matrix_Planu <- matrix(nrow = length(planu), ncol = length(planu))
colnames(corr_matrix_Planu) <- planu
rownames(corr_matrix_Planu) <- planu
pval_corr_matrix_Planu <- corr_matrix_Planu

corr_matrix_Meta <- matrix(nrow = length(meta), ncol = length(meta))
colnames(corr_matrix_Meta) <- meta
rownames(corr_matrix_Meta) <- meta
pval_corr_matrix_Meta <-corr_matrix_Meta

retention_matrix_Bila <- corr_matrix_Bila
retention_matrix_Planu <- corr_matrix_Planu
retention_matrix_Meta <- corr_matrix_Meta

for (taxon1 in taxons_vars)
  {
  for (taxon2 in taxons_vars)
    {
    if (taxon1 != taxon2){
      tmp_df <- df_Bila[,c('multi_sp', taxon1, taxon2)]
      tmp_df <- tmp_df %>% na.omit()
      len_df <- dplyr::tally(tmp_df)$n
      if (len_df > 0 ) {retention_matrix_Bila[taxon1, taxon2] <- len_df / 256}
      if (len_df > 10 ) {
        corr_matrix_Bila[taxon1, taxon2] <- cor.test(tmp_df[[taxon1]], tmp_df[[taxon2]], method = "spea
        pval_corr_matrix_Bila[taxon1, taxon2] <- cor.test(tmp_df[[taxon1]], tmp_df[[taxon2]], method =
          }
```

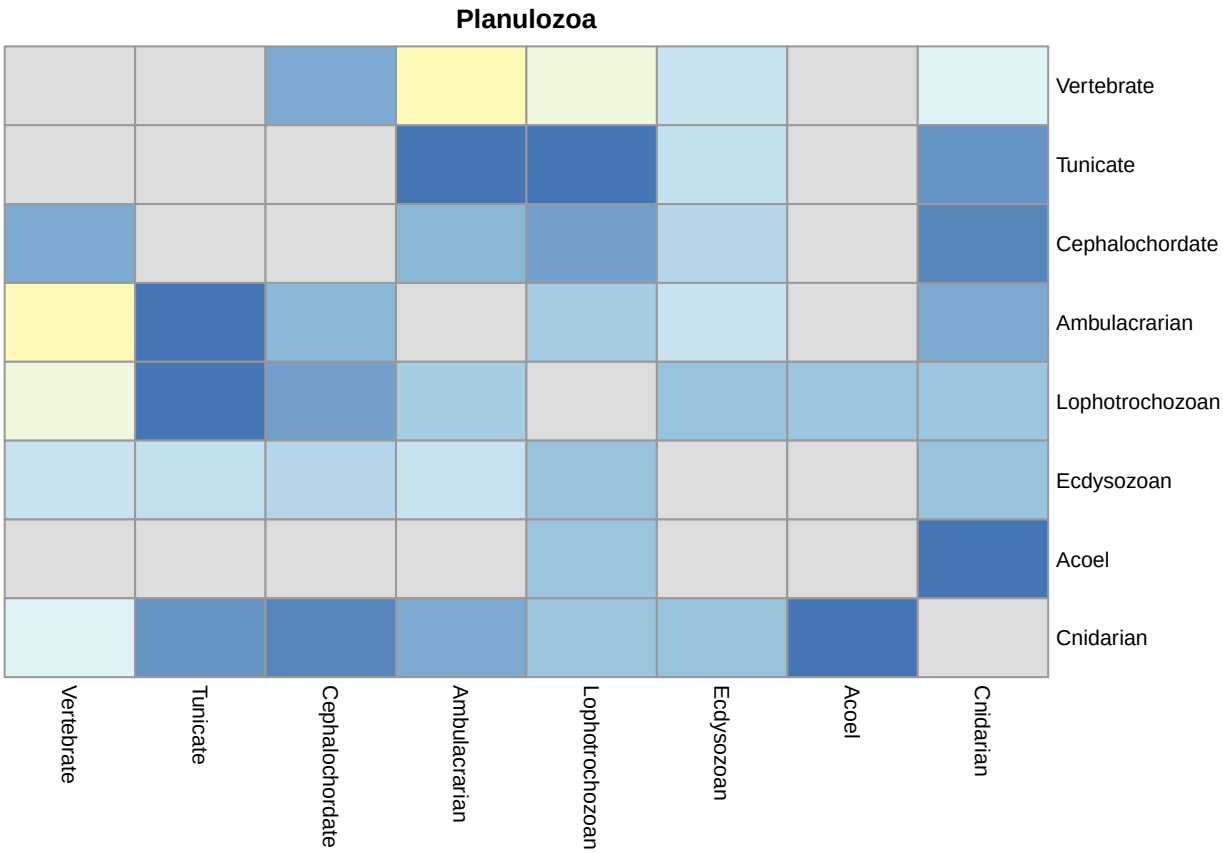
```
      tmp_df <- df_Planu[,c('multi_sp', taxon1, taxon2)]
      tmp_df <- tmp_df %>% na.omit()
      len_df <- length(tmp_df[[taxon1]])
      if (len_df > 0 ) {retention_matrix_Planu[taxon1, taxon2] <- len_df / 162}
      if (len_df > 10 ) {
        corr_matrix_Planu[taxon1, taxon2] <- cor.test(tmp_df[[taxon1]], tmp_df[[taxon2]], method = "spea
        pval_corr_matrix_Planu[taxon1, taxon2] <- cor.test(tmp_df[[taxon1]], tmp_df[[taxon2]], method =
      }

      tmp_df <- df_Meta[,c('multi_sp', taxon1, taxon2)]
      tmp_df <- tmp_df %>% na.omit()
      len_df <- length(tmp_df[[taxon1]])
      if (len_df > 0 ) {retention_matrix_Meta[taxon1, taxon2] <- len_df / 34}
      if (len_df > 10 ) {
        corr_matrix_Meta[taxon1, taxon2] <- cor.test(tmp_df[[taxon1]], tmp_df[[taxon2]], method = "spea
        pval_corr_matrix_Meta[taxon1, taxon2] <- cor.test(tmp_df[[taxon1]], tmp_df[[taxon2]], method =
      }
      }
  }
}
```

Heatmap of density correlation Also heatrmap of pairwise retention (supplement to scatterplots)

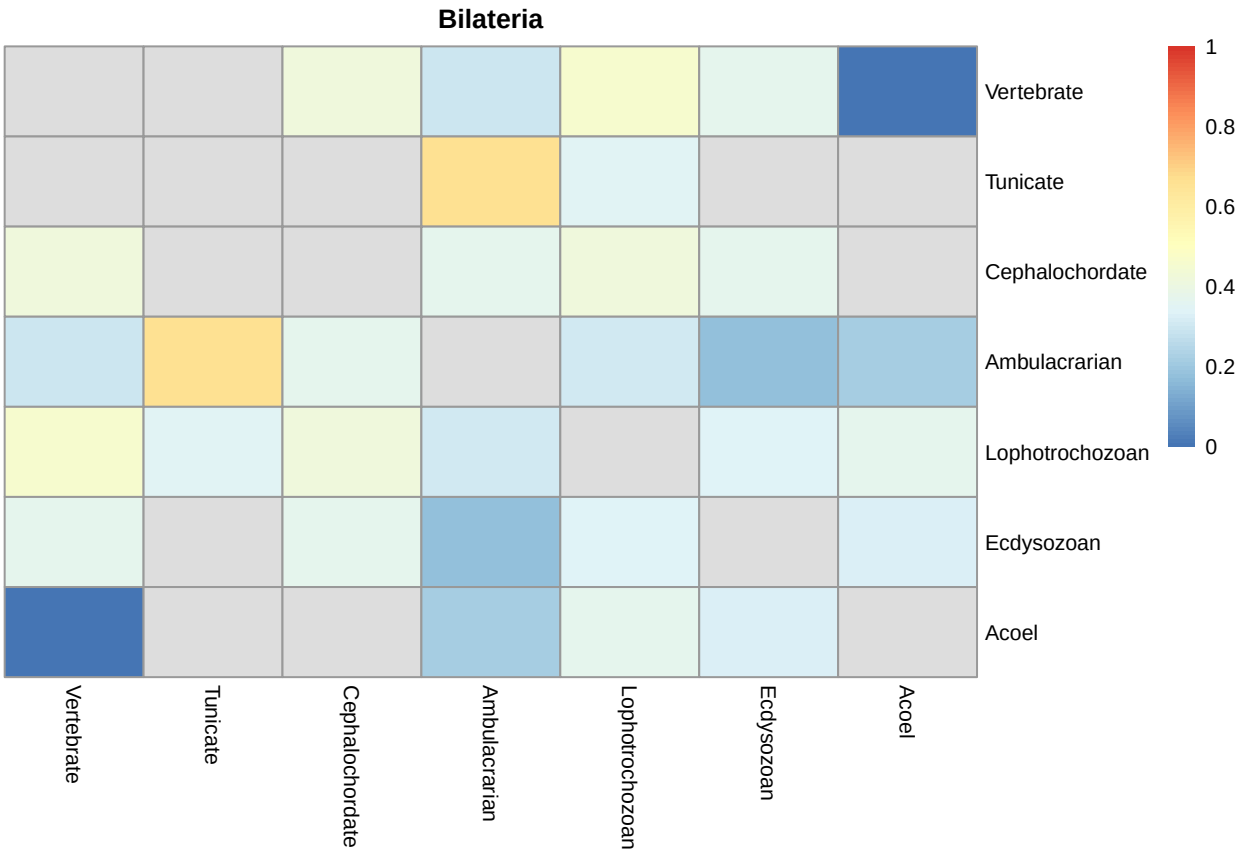
```
bk <- c(seq(0, 1,  length=100))

p1 <- pheatmap::pheatmap(corr_matrix_Meta,
                         cluster_rows = F,
                         cluster_cols = F,
                         breaks = bk,
                         legend = F,
                         fontsize = 8,
                         main = 'Metazoa')
```



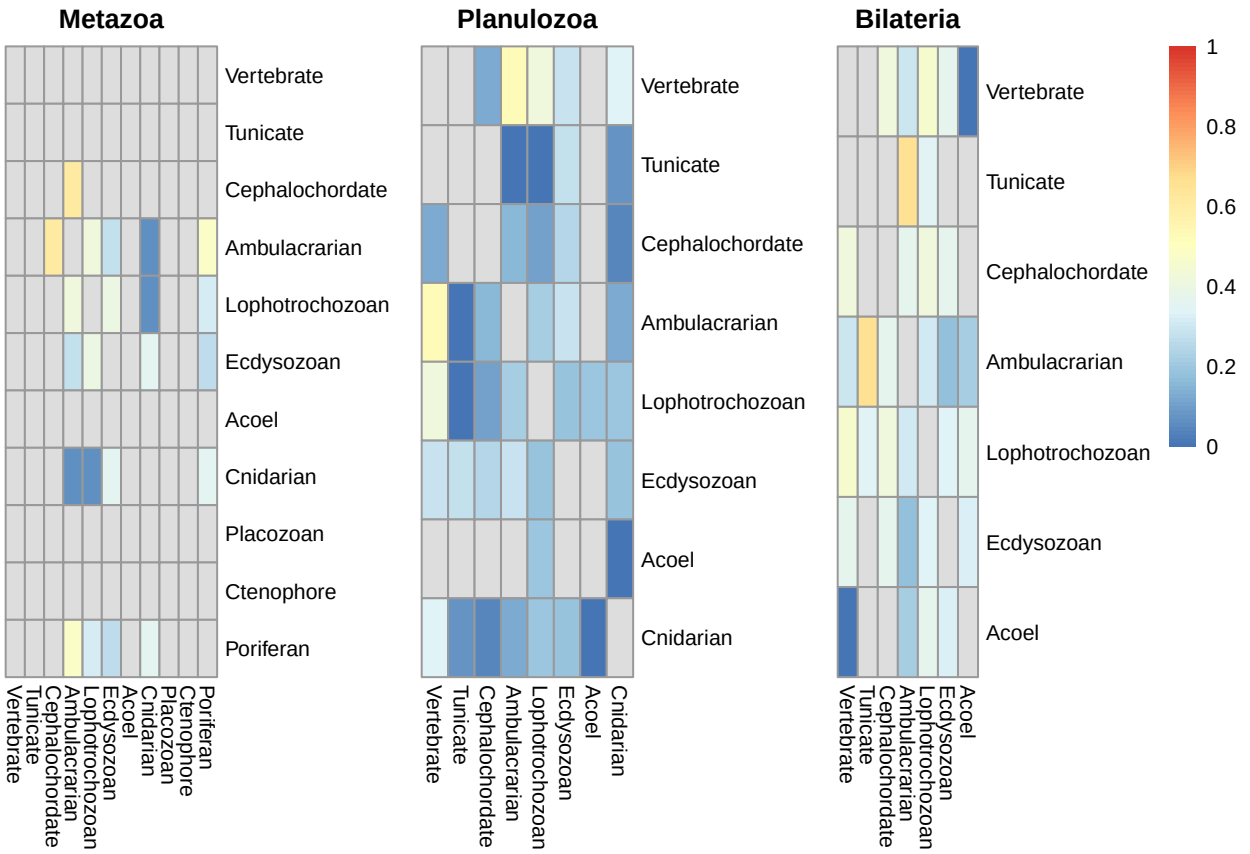

```
p2 <- pheatmap::pheatmap(corr_matrix_Planu,
                         cluster_rows = F,
                         cluster_cols = F,
                         breaks = bk,
                         legend = F,
                         fontsize = 8,
                         main = 'Planulozoa')
```




```
p3 <- pheatmap::pheatmap(corr_matrix_Bila,
                         cluster_rows = F,
                         cluster_cols = F,
                         breaks = bk,
                         fontsize = 8,
                         main = 'Bilateria')
```



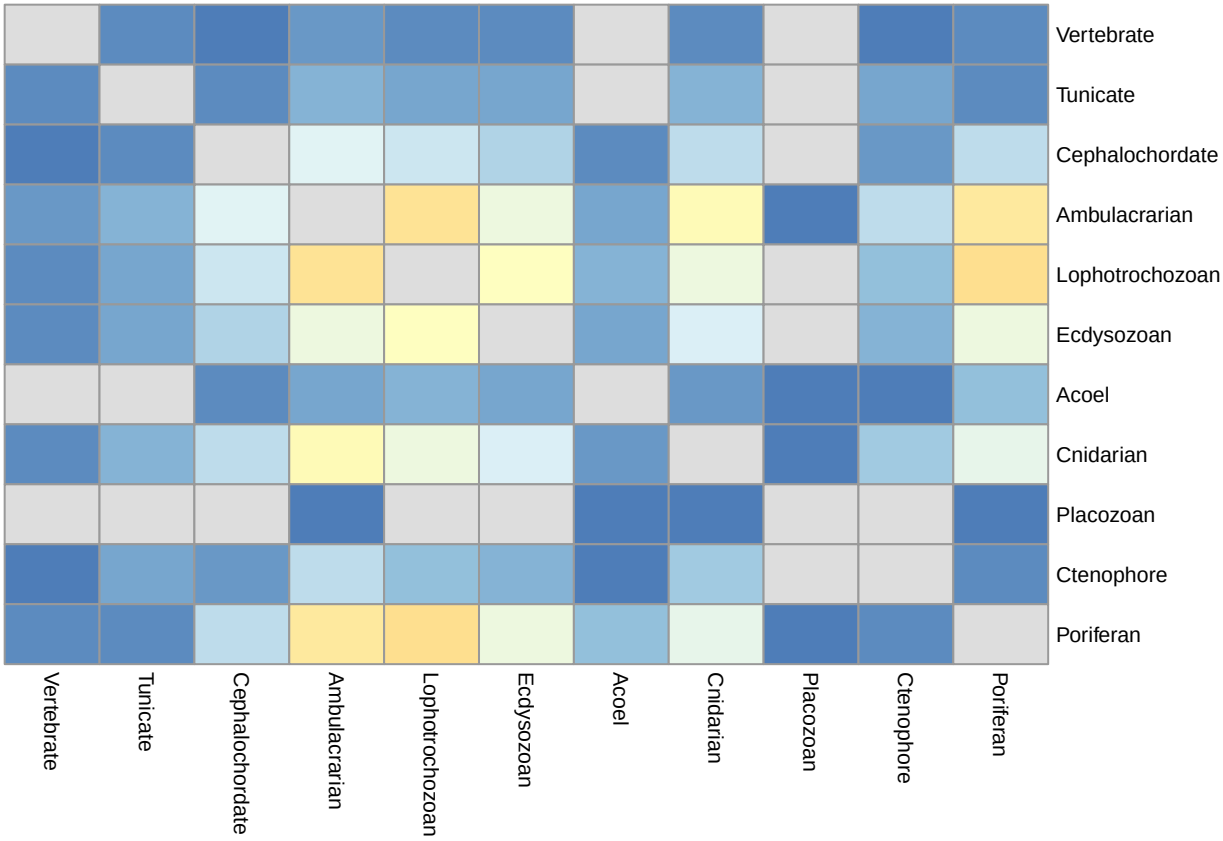

```
gridplot <- gridExtra::grid.arrange(grobs = list(p1$gtable,p2$gtable,p3$gtable), ncol = 3)
```



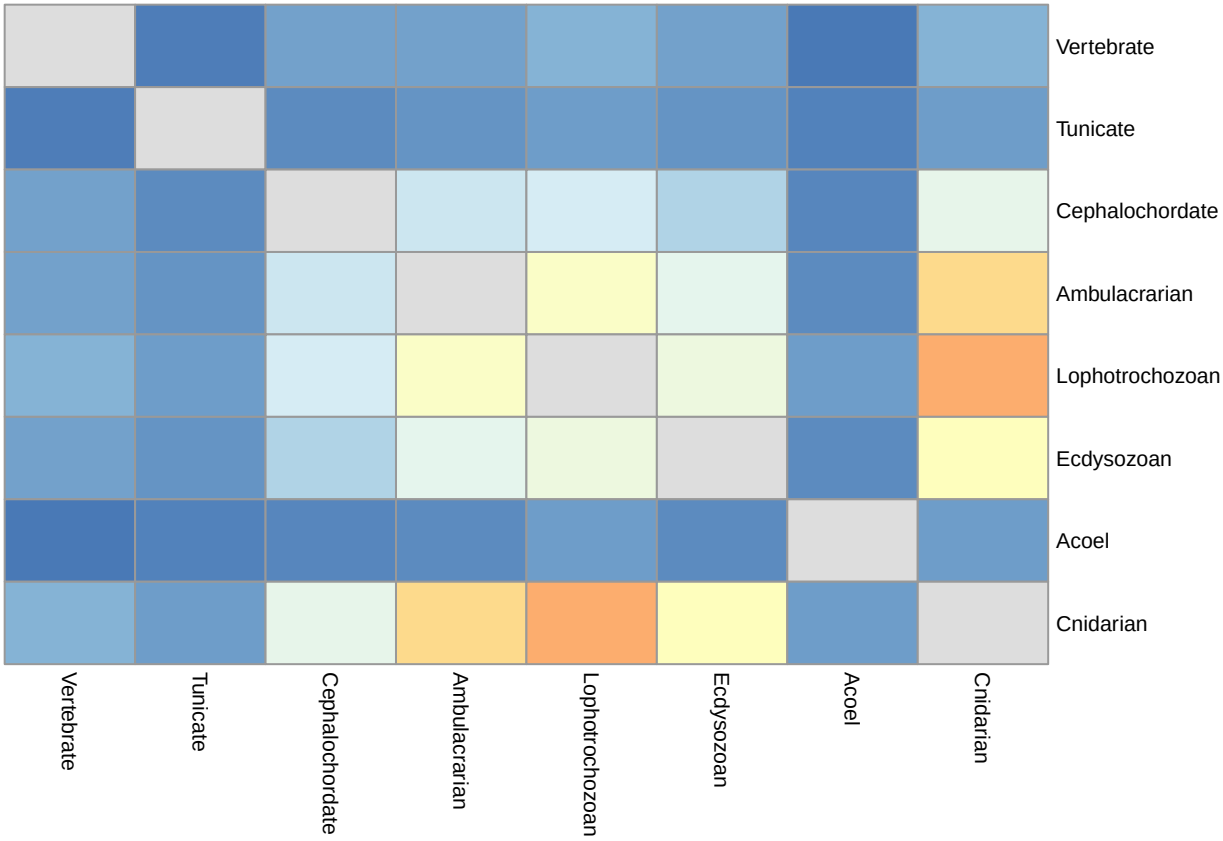

```
ggsave(plot = gridplot,
       filename = 'fig2_heatmaps_corr_density.pdf',
       unit = 'cm',
       width = 16,
       height = 6)
```

Now the pairwise retention matrix (SF5)

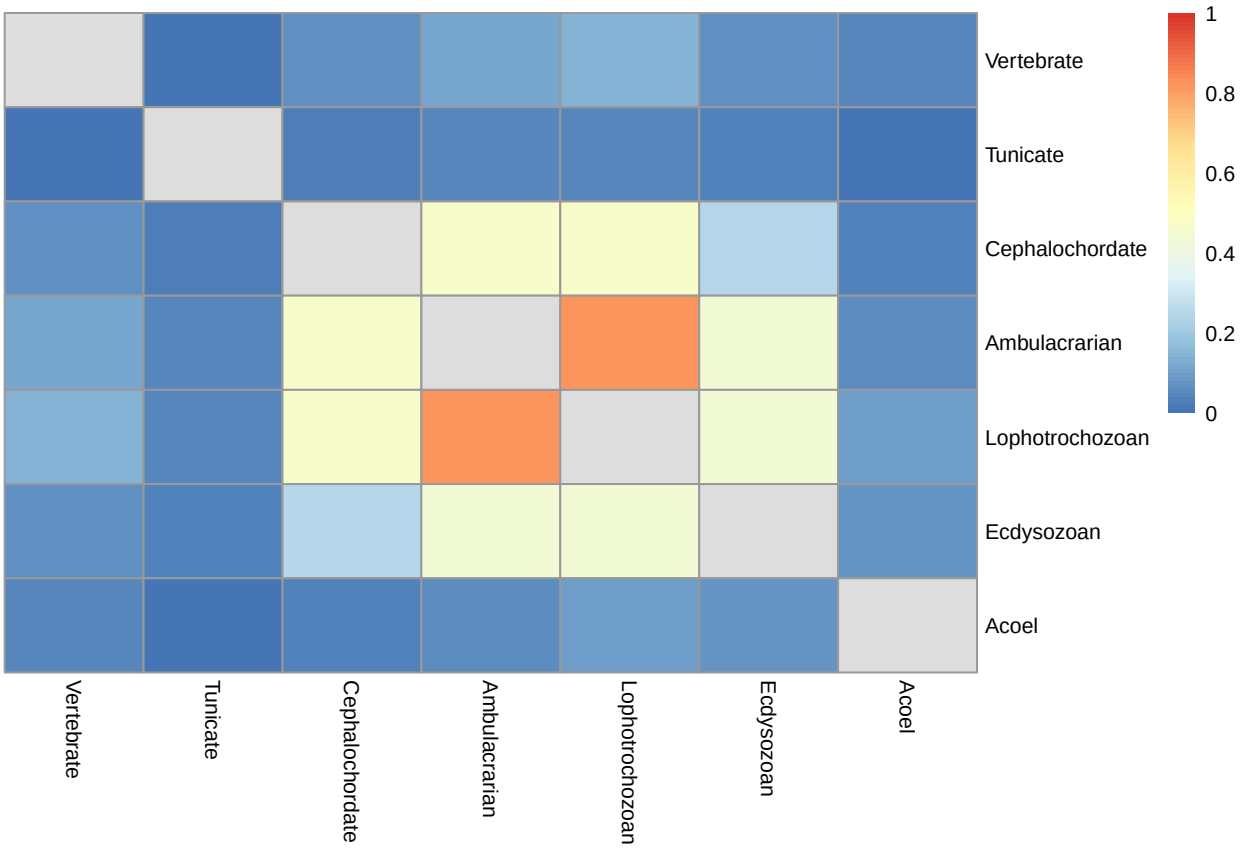
```
p4 <- pheatmap::pheatmap(retention_matrix_Meta,
                         cluster_rows = F,
                         cluster_cols = F,
                         breaks = bk,
                         legend = F,
                         fontsize = 8)
```



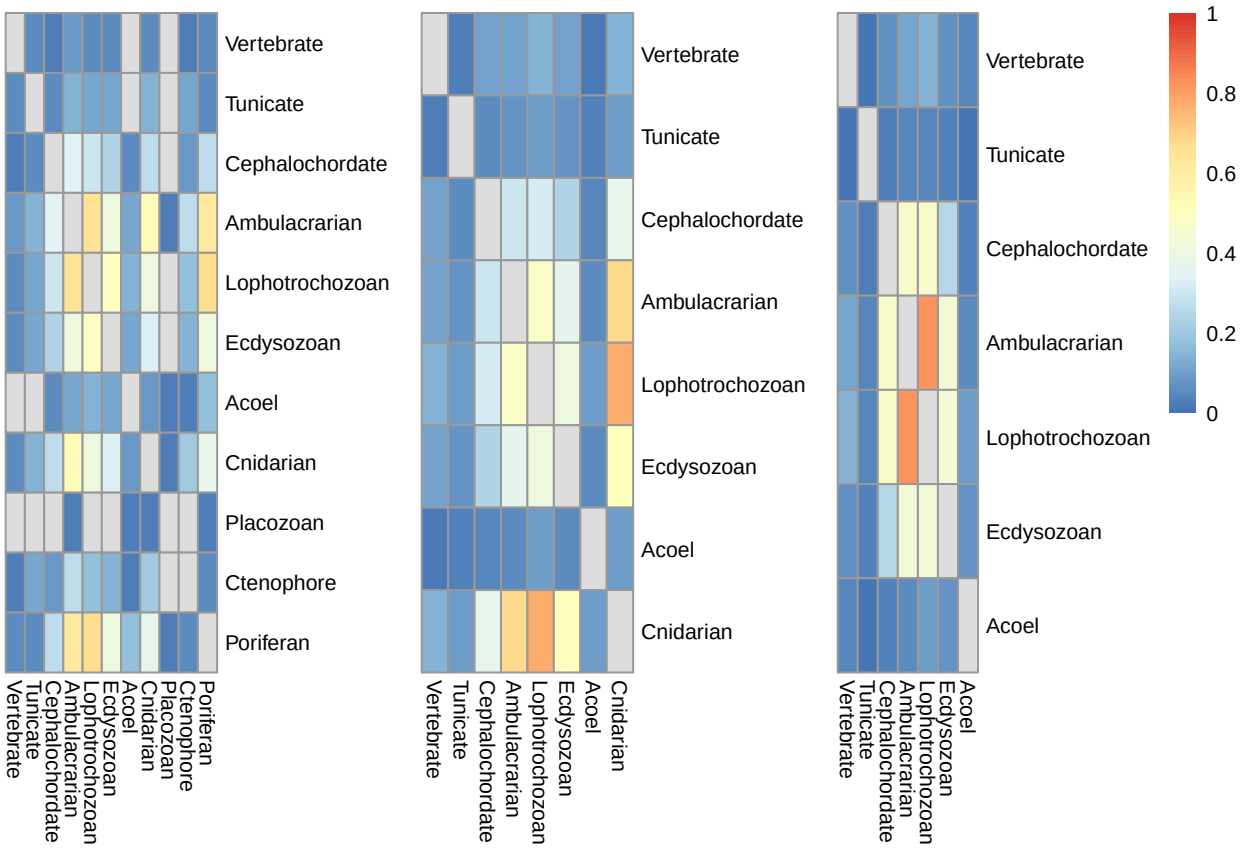

```
p5 <- pheatmap::pheatmap(retention_matrix_Planu,
                         cluster_rows = F,
                         cluster_cols = F,
                         breaks = bk,
                         legend = F,
                         fontsize = 8)
```




```
p6 <- pheatmap::pheatmap(retention_matrix_Bila,
                         cluster_rows = F,
                         cluster_cols = F,
                         breaks = bk,
                         fontsize = 8)
```




```
gridplot2 <- gridExtra::grid.arrange(grobs = list(p4$gtable,p5$gtable,p6$gtable), ncol = 3)
```




```
ggsave(plot = gridplot2,
       filename = 'SF6A_pairwise retention_corr_check.pdf',
       unit = 'cm',
       width = 16,
       height = 6)
```