

Grouped plots

Group plots

```
library(patchwork)
```

The plots for the paper need to be grouped into larger figures. The roughly planned grouping so far can be seen in the file `figure_group_planning.md`.

Load dependencies

```
library(ggplot2)
library(ontologyIndex)
library(ontologyPlot)
library(EWCE)
library(cowplot)
library(wesanderson)
library(dplyr)

data(hpo)
phenotype_to_genes = HPOExplorer::load_phenotype_to_genes("data/phenotype_to_genes.txt")
disease_descriptions = readRDS("data/disease_descriptions.Rda")
rownames(disease_descriptions) = disease_descriptions$HPO_id
load("data/Descartes_All_Results_extras.rda")
descartes_mappings = read.csv("data/DescartesHuman_celltype_mapping.csv")
descartes_mappings$level1 = gsub("_", " ", descartes_mappings$level1)
```

```

#tm_mappings = read.csv("data/TabulaMuris_celltype_mapping.csv")
ctd = readRDS("data/CTD_Descartes_withplot.rds")

source("source/cell_select_ggnetwork_plot.R")
source("source/phenotypes_per_cell_plot.R")
source("source/ewce_plot_function.R")

if (!require(HPOEWCE)) {
  #install.packages("rd_package/RareDiseaseEWCE", repos=NULL, type="source")
  devtools::install_github("ovrhuman/HPOEWCE")
  library(HPOEWCE)
}

if (!require(MultiEWCE)) {
  #install.packages("rd_package/RareDiseaseEWCE", repos=NULL, type="source")
  devtools::install_github("neurogenomics/MultiEWCE")
  library(MultiEWCE)
}

if (!require(HPOExplorer)) {
  #install.packages("rd_package/RareDiseaseEWCE", repos=NULL, type="source")
  devtools::install_github("neurogenomics/HPOExplorer")
  library(HPOExplorer)
}

```

ggplotify function

The facet packages (patchwork and cowplot facet_grid etc) dont seem to work with non-ggplot plots. So this function is here to convert the plots to ggplot format

```

savePlot <- function (plot, w = 1000, h = 1000, path = "savedPlot.png") {
  png(path,width = w, height = h)
  print(plot)
  dev.off()
}

```

```

}

ggloadImage <- function(path) {
  img <- png::readPNG(path)
  g <- grid::rasterGrob(img, interpolate = TRUE)
  plt <- qplot(1:10, 1:10, geom="blank") +
    annotation_custom(g, xmin=-Inf, xmax=Inf, ymin = -Inf, ymax=Inf) +
    theme_blank()
  return(plt)
}

ggplotify <- function(plot_object, height_px = 1000, width_px = 1000) {
  fp <- "ggplotify_temp.png"
  savePlot(plot_object, width_px, height_px, fp)
  plt <- ggloadImage(fp)
  file.remove(fp)
  return(plt)
}

# ggplotify <- function(plot_object, height_px = 1000, width_px = 1000) {
#   fp <- "ggplotify_temp.png"
#   savePlot(plot_object, width_px, height_px, fp)
#   img <- readPNG(fp)
#   file.remove(fp)
#   g <- rasterGrob(img, interpolate = TRUE)
#   plt <- qplot(1:10, 1:10, geom = "blank") +
#     annotation_custom(g, xmin=-Inf, xmax=Inf, ymin = -Inf, ymax=Inf) +
#     theme_blank()
#   return(plt)
# }

```

Figure 1 - Explanatory and introductory plots

```
# ADHD ancestor terms
namer = function(term){
  return (hpo$name[term])
}

terms = get_term_property(hpo,property="ancestors" ,term ="HP:0007018",
                          as_names=FALSE)

adhd_ancestors <- onto_plot(hpo, terms= terms[2:length(terms)],label = namer,
                           shape="circle",fontsize = 80, edge_attributes = list(color="grey"))

# APP Homepage screenshot
app_home = ggloadImage("figures/EWCE_home.png")

# Interactive app screenshot
app_interactive = ggloadImage("figures/interactive_cell_cellect_acinar_demo.png")

# Print friendly network plot
cell = "Acinar cells"

printable_networkPlot <- one_cell_ontology_plot_heatmap(all_results_merged,cell=cell, heatmapped_value = 1)
```

Here is an example of Figure 1

```
adhd_ancestors <- ggplotify(adhd_ancestors)
printable_networkPlot <- ggplotify(printable_networkPlot)

(adhd_ancestors | app_home) / (printable_networkPlot | app_interactive)
```

Figure 2 - Overview results

These figures are meant to give a general idea of how many results there are, and show that they make sense.

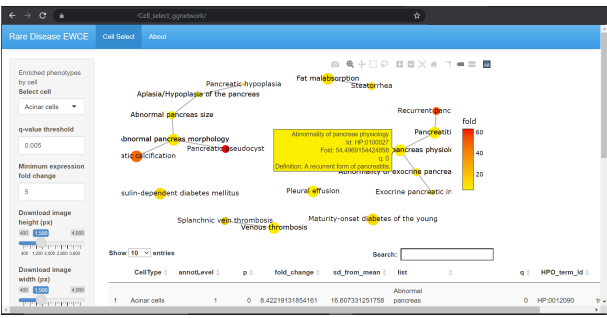
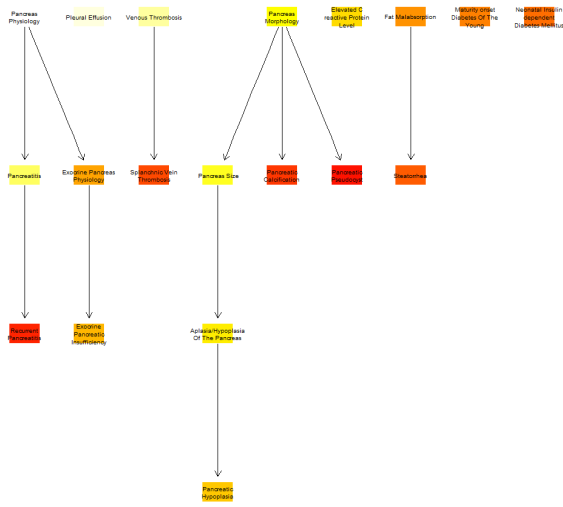
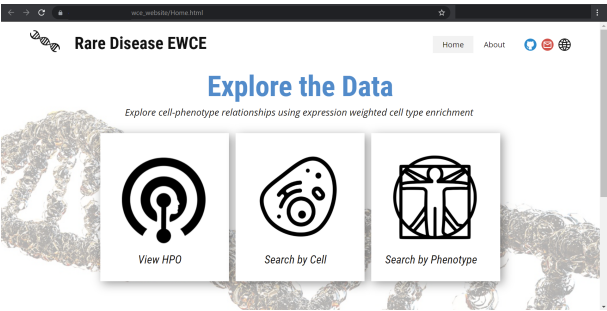


Figure 1: **Explanatory and Introductory plots.** These plots were mostly used in the introduction of my report.

```

# Phenos per cell plot
descartes_mappings = read.csv("data/DescartesHuman_celltype_mapping.csv")
descartes_mappings$level1 = gsub("_", " ", descartes_mappings$level1)
plot_npc = plot_phenos_per_cell1(all_results_merged, descartes_mappings, fold=1, q_val=0.05)

# Main branches of hpo
#main_branch_plt <- plot_n_phenotypes_per_branch_hpo(phenotype_to_genes=phenotype_to_genes, hpo=hpo)
phenosPerBranch <- function (phenotype_to_genes, hpo, highlighted_branches = c("Abnormality of the nervous system",
                                                                              "Abnormality of the cardiovascular system",
                                                                              "Abnormality of the digestive system"),
                             set_colors = c("#619CFF", "#F8766D", "#00BA38") ,
                             background_branches = hpo$children["HP:0000118"][[1]],
                             wes_anderson_palette = "Moonrise3")
{
  color_pal <- wesanderson::wes_palette(wes_anderson_palette,
                                         2)

  highlighted_branches_ids <- hpo$id[match(highlighted_branches,
                                           hpo$name)]

  phenos_per_branch <- data.frame()
  for (b in background_branches) {
    n <- length(ontologyIndex::get_descendants(hpo, b))
    if (b %in% highlighted_branches_ids) {
      target_branch <- "target"
    }
    else {
      target_branch <- "Other"
    }
    phenos_per_branch <- rbind(phenos_per_branch, data.frame(branch = hpo$name[b],
                                                            n_phenos = n, target = target_branch))
  }
  phenos_per_branch$branch <- stats::reorder(phenos_per_branch$branch,
                                             phenos_per_branch$n_phenos)

```

```

phenos_per_branch$color <- "gray"
for (i in seq(1, length(set_colors))) {
  phenos_per_branch$color[phenos_per_branch$branch == highlighted_branches[i]] <- set_colors[i]
}

phenos_per_branch_plt <- ggplot(phenos_per_branch, aes(x = n_phenos,
                                                    y = branch, color = target, fill = target)) +
  geom_segment(mapping = aes(xend = 0, yend = branch), size = 3, colour = phenos_per_branch$color) +
  ylab(element_blank()) +
  scale_color_manual(values = color_pal) +
  theme(axis.line.x = element_blank(), panel.background = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        axis.line.y = element_line(color = "black"),
        axis.ticks.x = element_blank(), axis.text.x = element_blank(),
        legend.position = "none")
return(phenos_per_branch_plt)
}

main_branch_plt <- phenosPerBranch(phenotype_to_genes, hpo)

# Phenos per cell by branch
plot_branches = c("Abnormality of the nervous system", "Abnormality of the cardiovascular system", "Abnormality of the digestive system")
facet_branch_plt = plot_n_signif_phenos_per_cell_by_branch(all_results_merged, plot_branches = plot_branches)

```

Here is Fig 2

```
print(plot_npc / ( main_branch_plt | facet_branch_plt ))
```

Figure 3 HPO Patterns and relationships

These show common patterns seen in HPO, related to oncology level, n genes etc.


```

# Significance and expected cell type #####
plot_branches = c("Abnormality of the nervous system", "Abnormality of the cardiovascular system", "Abn
expected_cells = c("Excitatory neurons", "Cardiomyocytes", "Antigen presenting cells")
correlation_results = data.frame()
proportion_plots = list()
for (i in seq(length(plot_branches))) {
  cur_plot <- proportion_of_expected_enrichments_plot(all_results_merged,
    hpo,
    target_cells=c(expected_cells[i]),
    cell_type_description = expected_cells[i],
    HPO_Ids = ontologyIndex::get_descendants(hpo,hpo$id[match(plot_branches[i],hpo$name)]),
    phenotype_description = plot_branches[i],
    wes_color_palette="Darjeeling1",
    n_colors = 4,
    color_expected_phenotypes = i,
    color_other_phenotypes = 4)
  proportion_plots[[i]] = cur_plot[[1]]
  correlation_results = rbind(correlation_results,cur_plot[[2]])
}

sigplot <- cowplot::plot_grid(plotlist=proportion_plots, align = "h",nrow = 1,labels=c("A","B","C"))

# ont level plots (violin) #####

ont_levels <- data.frame("phenotype"=unique(phenotype_to_genes$Phenotype),
                        "hpo_id"=hpo$id[match(unique(phenotype_to_genes$Phenotype),hpo$name)])
ont_levels <- ont_levels[complete.cases(ont_levels),]
lvls <- c()
for (id in ont_levels$hpo_id) {
  lvls = append(lvls, get_ont_level(hpo,id))
}

```

```

ont_levels$ont_lev <- lvls
rm(lvls)

n_associated_cells <- c()
for (p in ont_levels$phenotype) {
  #n_associated_cells <- append(n_associated_cells, length(all_results_merged$CellType[all_results_merg
  n_associated_cells <- append(n_associated_cells, length(all_results_merged$CellType[all_results_merg
}
ont_levels$n_associated_cells <- n_associated_cells
rm(n_associated_cells)

pal <- wesanderson::wes_palette("Darjeeling2", n=2)
ontlvl_ncells_plt <- ggplot(ont_levels, mapping=aes(x= factor(ont_lev), y=n_associated_cells)) +
  geom_jitter(color = pal[1]) +
  geom_violin(fill = NA) +
  geom_smooth(color = pal[2], method="loess", mapping = aes(x=ont_lev)) +
  cowplot::theme_cowplot() +
  labs(x="Ontology level", y="Associated cells/phenotype (n)", title = "n associated cells/phenotype by c
#ontlvl_ncells_plt

# ont level facet

signif_res <- all_results_merged[all_results_merged$q < 0.05,]
ontlevz <- c()
for (p in unique(signif_res$HPO_id)) {
  ontlevz[p] <- get_ont_level(hpo,p)
}
signif_res$ontlvl <- ontlevz[signif_res$HPO_id]
signif_res <- signif_res[complete.cases(signif_res),]

```

```

pal <- wesanderson::wes_palette("Darjeeling2", n=2)
ontlvl_fold_plt <- ggplot(signif_res, aes(x = factor(ontlvl), y = fold_change)) +
  geom_jitter(color = pal[1]) +
  geom_violin(fill = NA) +
  geom_smooth(color = pal[2], method="loess", mapping = aes(x=ontlvl)) +
  cowplot::theme_cowplot() +
  labs(x="Ontology level", y="Fold change", title = "Fold change in specific expression by ontology level")

# ngenes plt (stat smooth takes too long/doesn't work with all results so just use signif ?)
ngenes<-c()
for(p in unique(signif_res$list)) {
  ngenes[p] <- length(get_gene_list(p, phenotype_to_genes))
}
signif_res$ngenes <- ngenes[signif_res$list]
signif_res <- signif_res[complete.cases(signif_res),]
pal <- wesanderson::wes_palette("Darjeeling2", n=2)
ontlvl_ngenes_plt <- ggplot(signif_res, aes(x = factor(ontlvl), y = ngenes)) +
  geom_jitter(color = pal[1]) +
  geom_violin(fill = NA) +
  geom_smooth(color = pal[2], method="loess", mapping = aes(x=ontlvl)) +
  cowplot::theme_cowplot() +
  labs(x="Ontology level", y="Genes (n)", title = "Number of genes by ontology level")

ontlvlplots <- cowplot::plot_grid(plotlist=list(ontlvl_ncells_plt,ontlvl_fold_plt,ontlvl_ngenes_plt),align="h")

```

Here is Fig 3

```
print(sigplot / ontlvlplots )
```

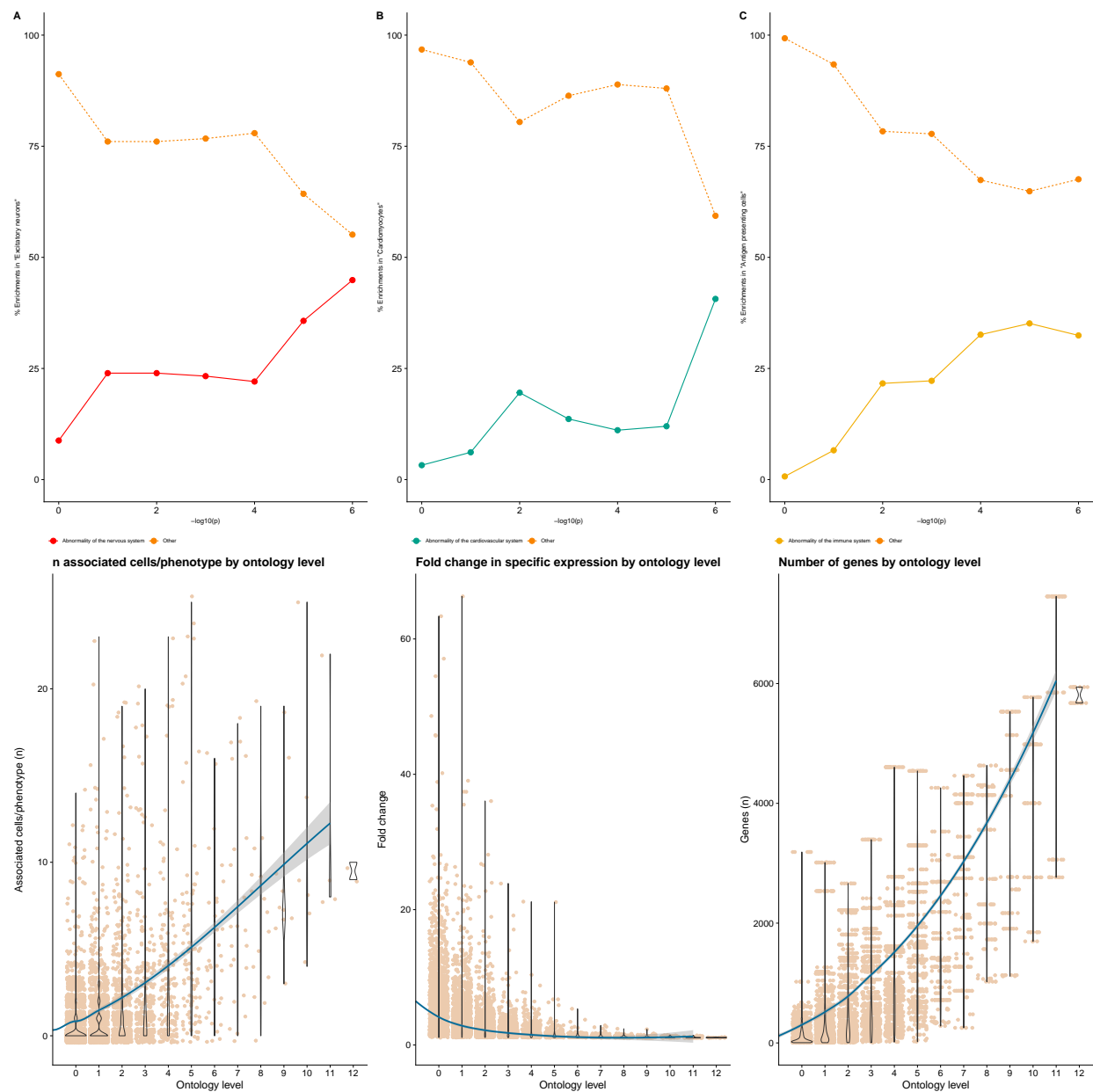


Figure 3: **ontology level plots** Plots describing relationships between expected cell types, significance, and ontology level.

Figure 4 infections

```
namer = function(term){
  return (hpo$name[term])
}

terms = ontologyIndex::get_descendants(hpo,"HP:0002719", exclude_roots = FALSE)
infect_ancestors <- onto_plot(hpo, terms= terms[2:length(terms)],label = namer,
  shape="circle",fontsize = 80, edge_attributes = list(color="grey"))
infect_ancestors <- ggplotify(infect_ancestors)
# branch_plt #####
library(wesanderson)
color_pal = wes_palette("Darjeeling1",4)
library(cowplot)

branch = "Recurrent infections"
branch_id = paste(hpo$id[match(branch,hpo$name)])
branch_descendants = paste(get_descendants(hpo,branch_id))
branch_descendants_names = paste(hpo$name[branch_descendants])
all_results_merged$list = paste(all_results_merged$list)
#all_results_merged$cur_branch = paste(all_results_merged$cur_branch)

all_results_merged$cur_branch = paste("Other")
all_results_merged$cur_branch[all_results_merged$list %in% branch_descendants_names] = branch

branch_signif_counts = data.frame()
for (c in unique(all_results_merged$CellType)) {
  n_signif = length(all_results_merged[all_results_merged$CellType == c & all_results_merged$cur_branch
  branch_signif_counts = rbind(branch_signif_counts,
    data.frame("branch"=branch,
```

```

        "CellType"=c,
        "n_signif"=n_signif))
}

cell_order = factor(gsub("_"," ",ctd[[1]]$plotting$cell_ordering))
branch_signif_counts$cell_order = match(branch_signif_counts$CellType, cell_order)
branch_signif_counts$CellType = reorder(branch_signif_counts$CellType, branch_signif_counts$cell_order)
branch_signif_counts$labels = branch_signif_counts$n_signif
branch_signif_counts$labels[branch_signif_counts$labels == 0] = ""

branch_plt <- ggplot(branch_signif_counts[branch_signif_counts$branch==branch,], aes(x=CellType,y=n_signif)) +
  geom_col( fill = color_pal[2], color = "black") +
  geom_text(mapping= aes(label = labels, y = n_signif + 3))+
  theme_cowplot()+
  ylab("N phenotypes") +
  scale_y_continuous(expand=c(0,0), limits= c(0,max(branch_signif_counts$n_signif)+5))+
  theme(axis.text.x = element_text(angle = 90, hjust=1, vjust = 0.2),legend.position="none") +
  #coord_flip() +
  ggtitle("Significant enrichments per cell: Recurrent infections")

#branch_plt

# pheno_fold_plt #####

branch = "Recurrent bacterial infections"
exclude_root = TRUE

branch_id = paste(hpo$id[match(branch,hpo$name)])
branch_descendants = hpo$children[[branch_id]]
branch_descendants_names = paste(hpo$name[branch_descendants])
if (exclude_root) {

```

```

    branch_descendants_names = branch_descendants_names[branch_descendants_names != branch]
  } else {
    branch_descendants_names = c(branch, branch_descendants_names)
  }

pheno_df = all_results_merged[all_results_merged$list %in% branch_descendants_names, ]

pheno_df$signif_asterics = ""
pheno_df$signif_asterics[pheno_df$q<0.05] = "*"
pheno_df$signif_asterics[pheno_df$q<0.001] = "***"
pheno_df$signif_asterics[pheno_df$q<0.0001] = "****"
pheno_df$signif_asterics[pheno_df$q<0.00001] = "*****"

recurrentBact_fold_plt <- ggplot(pheno_df, aes(x = CellType, y= fold_change, fill = list)) +
  ggtitle(paste0('Child nodes of HP0 branch "',branch,'"')) +
  geom_col() +
  geom_text(label = pheno_df$signif_asterics, mapping = aes(y = fold_change + 1))+
  theme_cowplot() +
  scale_y_continuous(expand=c(0,0), limits= c(0,max(branch_signif_counts$n_signif)+2))+
  ylab("Fold change") +
  xlab("Cell type") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.2),legend.position = "none") +
  facet_wrap(~list, ncol=1)

#####

branch = "Recurrent gram-negative bacterial infections"
exclude_root = TRUE

branch_id = paste(hpo$id[match(branch,hpo$name)])
branch_descendants = hpo$children[[branch_id]]
branch_descendants_names = paste(hpo$name[branch_descendants])
if (exclude_root) {
  branch_descendants_names = branch_descendants_names[branch_descendants_names != branch]
}

```

```

} else {
  branch_descendants_names = c(branch, branch_descendants_names)
}

pheno_df = all_results_merged[all_results_merged$list %in% branch_descendants_names, ]

pheno_df$signif_asterics = ""
pheno_df$signif_asterics[pheno_df$q<0.05] = "*"
pheno_df$signif_asterics[pheno_df$q<0.001] = "***"
pheno_df$signif_asterics[pheno_df$q<0.0001] = "****"
pheno_df$signif_asterics[pheno_df$q<0.00001] = "*****"

recurrentGram_plt <- ggplot(pheno_df, aes(x = CellType, y= fold_change, fill = list)) +
  ggtitle(paste0('Child nodes of HPO branch "',branch,'"')) +
  geom_col() +
  geom_text(label = pheno_df$signif_asterics, mapping = aes(y = fold_change + 1))+
  theme_cowplot() +
  scale_y_continuous(expand=c(0,0), limits= c(0,max(branch_signif_counts$n_signif)+2))+
  ylab("Fold change") +
  xlab("Cell type") +
  theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.2),legend.position = "none") +
  facet_wrap(~list, ncol=1)

```

Here is Fig 4

```
print((branch_plt|infect_ancestors)/(recurrentBact_fold_plt|recurrentGram_plt))
```

Figure - social interactions

```

library(EWCE)
library(ggplot2)
library(cowplot)

```



```

source("source/ewce_plot_function.R")

# Dendrogram and signif cell types
pheno = "Impaired social interactions"
subset_results = all_results_merged[all_results_merged$list == pheno, ]
subset_results$CellType = gsub(" ", "_", subset_results$CellType)
plt1 = ewce.plot(subset_results, mtc_method = "BH", ctd = ctd)$withDendro

```

```

## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.

```

```

# social interactions
namer = function(term){
  return (hpo$name[term])
}

terms = ontologyIndex::get_descendants(hpo, "HP:0000735", exclude_roots = FALSE)
terms = append(terms, "HP:0000735")
social_ancestors <- onto_plot(hpo, terms= terms[2:length(terms)], label = namer,
                             shape="circle", fontsize = 80, edge_attributes = list(color="grey"))
social_ancestors <- ggplotify(social_ancestors)

```

```

## poor eye contact

```

```

library(EWCE)
library(ggplot2)
library(cowplot)
source("source/ewce_plot_function.R")

# Dendrogram and signif cell types
pheno = "Poor eye contact"
subset_results = all_results_merged[all_results_merged$list == pheno, ]
subset_results$CellType = gsub(" ", "_", subset_results$CellType)

```

```
plt2 = ewce.plot(subset_results, mtc_method = "BH", ctd = ctd)$withDendro
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will  
## replace the existing scale.
```

Here is Fig ??

```
print(plt1/ (social_ancestors | plt2))
```

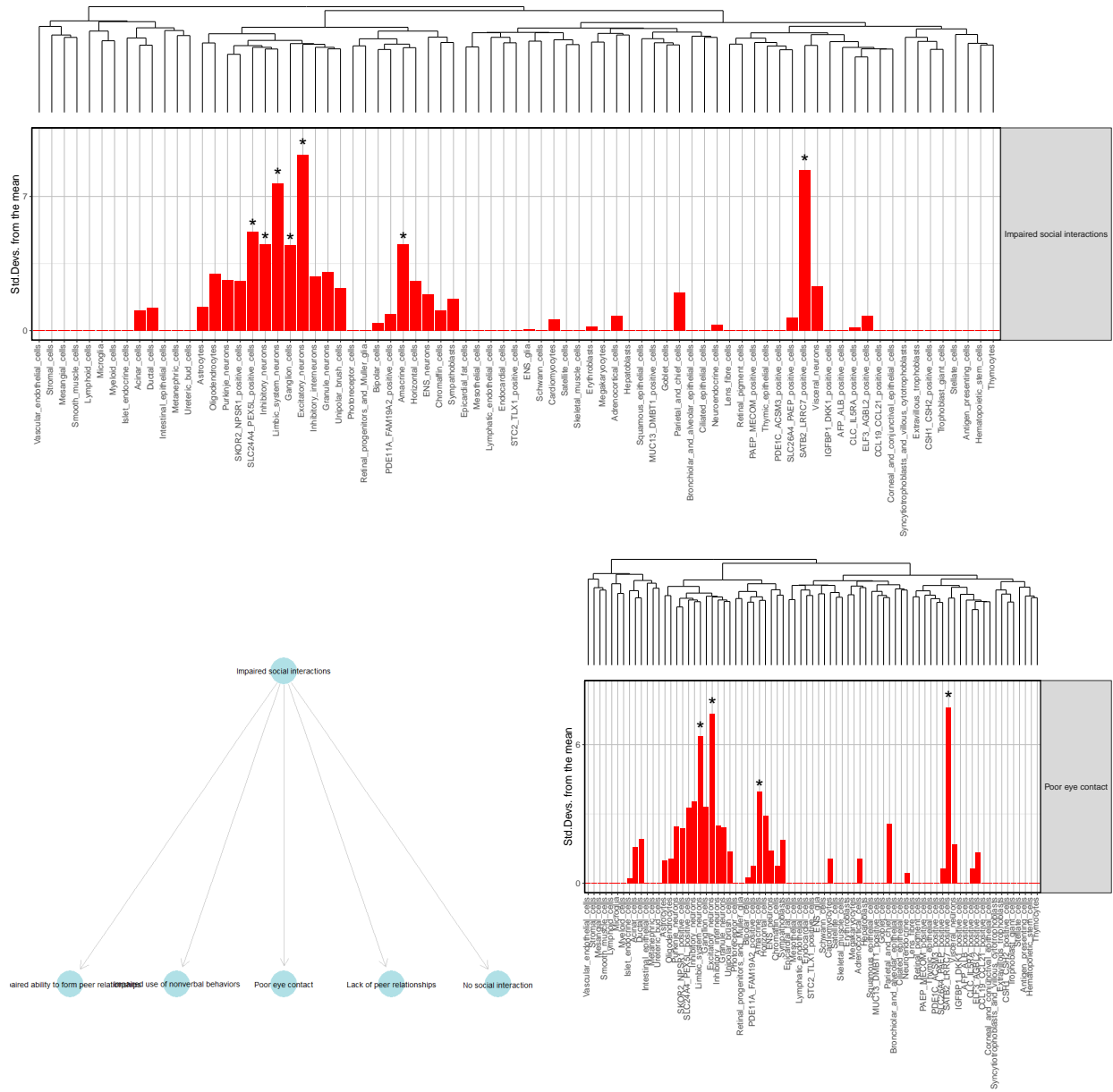


Figure 5: **Social interactions** Enrichments in cell types for impaired social interactions.