R codes of analysis serum (mice / rat) (neg)

Contents

1 File: serum_neg_stat_mcnebula.R

```
## gather sirius. a number of compounds candidate were collate
## and further annatate with classification
## here, classyfireR is used to add annotation of classification
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
candidates <- read_tsv("indo.tsv") %>%
  dplyr::mutate(sp.id = rownames(.)) %>%
  dplyr::as_tibble()
## -------------------------------------
## search pubchem
mutate_inchi_curl(candidates$inchikey2D, candidates$sp.id)
## gather data
pub_instance <- gather_inchi_curl() %>%
  dplyr::rename(.id = sp.id)
## ------------------
mutate_auto_classy(pub_instance, cl = 20)
## gather classyfire
simp_candi <- meta_gather_pub_classyfire_sirius(pub_instance, "Indoles and derivatives", candidates)
## ------------------------------------------------------------------------
system("rm -r inchi_pub")
```

```
## re-collate compound via pubchem
mutate_inchi_curl(simp_candi$inchikey2D, simp_candi$sp.id, get = "IUPACName")
system("rm -r inchi_pub")
## gather
name_df <- gather_inchi_curl() %>%
  dplyr::distinct(sp.id, .keep_all = T)
## as export
indo_export <- meta_re_collate_iupac_via_inchi(simp_candi, name_df, export)
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## mannualy adjust name >>> syno_backup
gt_indo_export <- indo_export %>%
  pretty_table(spanner = T, shorter_name = F, default = T)
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## the manipulation across through classyfire. therefore if databse without
## the compounds, the classification annotation possibly be leaving out
## this step re-search some compouns
mutate_inchi_curl_syno(candidates$inchikey2D, candidates$sp.id)
```

2 File: serum_neg_stat_metabo_collate.R

```
metabo_results <- metabo_collate(path = "~/Desktop")
## ------------------------------------
mutate_mz_rt <- dplyr::mutate(mz_rt, rt = rt * 60)
## get id
metabos <- metabo_get_id_via_mz_rt(metabo_results, mutate_mz_rt)
## gather export with metabos
mutate_export <- lapply(metabos, merge, y = export, by = "id", all.x = T) %>%
  lapply(dplyr::arrange, name) %>%
  lapply(dplyr::as_tibble) %>%
  data.table::rbindlist()
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## as pdf table
gt_export <- mutate_export %>%
  # mutate(name = "X") %>%
  dplyr::relocate(id, name, vip) %>%
  pretty_table(spanner = T, shorter_name = F, default = T)
```

```
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## draw enrichment results
## -------------------------------------
## pathway significant
metabo_pathway <- data.table::rbindlist(metabo_results) %>%
  dplyr::distinct(pathway, Hits.sig, Gamma)
pathway_horizon(metabo_pathway, title = "pathway enrichment")
```

3 File: serum__neg__stat__metabo.R

```
library(MetaboAnalystR)
## ------------------
p_col <- c("pro_model#q_value", "pro_model#log2.fc")
## ------------------
detach("package:dplyr")
# metabo_idenfication <- meta_metabo_pathway(export, mz_rt, p_col = p_col, ppm = 10, p_cutoff = 0.05, d
subm_list <- meta_metabo_pathway(export, mz_rt, p_col = p_col, only_return = T)
## the submit file is got and name as "tmp.txt"
## -------------------------------------
## collate file that download from Web of MetaboAnalyst
```

4 File: serum__neg__stat__metdna.R

```
## ------------------------------------------------------------------------
## reformat peak list and upload to metDNA
metdna_feature <- feature_csv %>%
  dplyr::select(contains("ID"), contains("m/z"), contains("retention"), contains("Peak area"))
## rename as metDNA recogonized name
colnames(metdna_feature) <- colnames(metdna_feature) %>%
  ## find and sort as order of `key`
  .meta_find_and_sort(., c("ID", "m/z", "retention")) %>%
  ## rename columns of df_mz_rt
  mapply_rename_col(., c("name", "mz", "rt"), colnames(metdna_feature)) %>%
  ## remove ` Peak area`
  gsub(" Peak area", "", .)
## -------------------------------------
metdna_feature <- metdna_feature %>%
  dplyr::mutate(rt = 60 * rt)
```

```
## ------------------------------------
write.csv(metdna_feature, file = "metdna_feature.csv", row.names = F, quote = F)
## ------------------------------------------------------------------------
## reformat
metdna_metadata <- metadata %>%
  dplyr::select(sample, group) %>%
  dplyr::rename(sample.name = sample)
## ------------------------------------
write.csv(metdna_metadata, file = "metdna_metadata.csv", row.names = F, quote = F)
## ------------------------------------------------------------------------
```

5 File: serum_neg_stat_ms1_search.R

```
## some specific structure need to search
comple <- read_tsv("indo_and_phenol.tsv") %>%
  dplyr::mutate(inchikey2D = stringr::str_extract(inchi, "^[A-Z]{1,1000}"))
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
# ## phenol structure candidates
# phenol <- read_tsv("phenol.tsv") %>%
#   dplyr::as_tibble()
# ## inchikey2D search
# search_results <- lapply(comple$inchikey2D, inchikey2d_search,
#                          db = candidates)
# inchikey2d_search(inchikey2D, db, col = "inchikey2D")
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
comple_formula <- inchikey_get_formula(comple$inchi) %>%
  dplyr::distinct(inchikey, .keep_all = T)
## get possibly precursor
precursor <- formula_adduct_mass(comple_formula$MolecularFormula)
names(precursor) <-  comple$name
## ------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## precursor search
sig_mz_rt <- dplyr::mutate(mz_rt, sig = ifelse(id %in% export$id, T, F))
```

```
mz_search <- multi_formula_adduct_align(precursor, sig_mz_rt, mz_tol = 0.005) %>%
  data.table::rbindlist(idcol = T, fill = T) %>%
  dplyr::rename(info = .id) %>%
  dplyr::mutate(id = as.character(id))
## ------------------------------------
much_export <- meta_summarise %>%
  meta_compound_filter(vip = vip, dose = "high",
                       l_abs_log_fc = 0, l_q_value = 1, l_vip = 0)
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
mz_search_export <- merge(mz_search, much_export, by = "id", all.x = T) %>%
  dplyr::filter(is.na(vip) == F)
## ------------------------------------
gt_mz_search_export <- mz_search_export %>%
  pretty_table(spanner = T, shorter_name = F, default = T, title = "Arachidonic acid compound search",
               subtitle = paste0("All search is >>> ", paste(comple$name, collapse = " | ")))
```

6 File: serum_neg_stat.R

```
## sample name annotation
GROUP <- c(blank_ = "blank", C = "control", M = "model", P = "positive",
           SL = "raw_low", SM = "raw_medium", SH = "raw_high",
           YL = "pro_low", YM = "pro_medium", YH = "pro_high")
## ------------------------------------------------------------------------
## set palette
palette <- c(control = "grey", model = "#374E55FF", drug = "#00A087FF",
             ## red for pro group
             pro_low = "#FDAE6BFF", pro_medium = "#FD8D3CFF", pro_high = "#E6550DFF",
             ## blue for raw group
             raw_low = "#9ECAE1FF", raw_medium = "#6BAED6FF", raw_high = "#3182BDFF")
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## R codes for meta analyses
# file <- "./batch_serum_neg.csv"
feature_csv <- data.table::fread(file) %>%
  dplyr::as_tibble()
## ------------------------------------------------------------------------
## gather peak_area
```

```
peak_area <- feature_csv %>%
  dplyr::select(contains("ID"), contains("Peak area"))
## ------------------------------------
mz_rt <- dplyr::select(feature_csv, 1:3)
colnames(mz_rt) <- c("id", "mz", "rt")
## -----------------
mz_rt <- mz_rt %>%
  dplyr::mutate(id = as.character(id))
## ------------------------------------
## collate metadata
metadata <- colnames(peak_area)[2:ncol(peak_area)] %>%
  meta_get_metadata()
## ------------------------------------
db_get_sample <- meta_do_list(metadata)
## all sample name and group and super group have been collate
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## in order to get that matrix
# |multi |multi |multi |cp   |cp   |cp   |
# |:-----|:-----|:-----|:----|:----|:----|
# |c1    |c1    |c1    |base |base |base |
# |c1    |c1    |c1    |base |base |base |
# |c1    |c1    |c1    |base |base |base |
# |c2    |c2    |c2    |c3   |c3   |c3   |
## ------------------------------------
## all compare couple
## base part
compare <- metadata$group %>%
  meta_get_couple()
## ------------------------------------
extra_compare <- compare %>%
  meta_get_extra_couple()
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## to get facet pca plot
## pca analyses
compute_df <- as_pca_df(peak_area)
## calculate
pca_set <- pca_via_group(df = compute_df, compare = compare,
```

```
                        extra_compare = extra_compare,
                        db_get_sample = db_get_sample)
## -------------------------------------
## visualize pca
plot <- visualize_facet_pca(pca_set, palette, metadata)
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## to get VIP value
## opls-da analyses
## only compare control with model
opls_set <- meta_oplsda(compute_df, metadata, c("control", "model"))
## get vip dataset
vip <- opls_set$vip
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## trans df format
mutate_peak_area <- compute_df %>%
  meta_array_to_df(., metadata)
## -------------------------------------
## calculate log2FC, p-value, q-value
meta_summarise <- meta_summarise_via_group(df = mutate_peak_area, compare = compare)
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## do specific dosage dispose
export <- meta_summarise %>%
  meta_compound_filter(vip = vip, dose = "high",
                       l_abs_log_fc = 1, l_q_value = 0.05)
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## module metabo analysis
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## model MCnebula cluster
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## MS 1 search
```

```
## ------------------------------------------------------------------
## ------------------------------------------------------------------
## ------------------------------------------------------------------
## module metDNA upload
## ------------------------------------------------------------------
## ------------------------------------------------------------------
## ------------------------------------------------------------------
```