R codes of analysis eucommia (previous)

Contents

1 File: bio_annotate_nebula.R

```
## palette
stat_palette <- c(Blank = "#B8B8B8",
                  `EU-Raw` = "#C6DBEF",
                  `EU-Pro` = "#FDD0A2")
## ------------------------------------------------------------------------
## show statistic legend for nebula
m.stat_palette <- stat_palette
names(m.stat_palette) <- c("Blank", "Raw-Eucommia", "Pro-Eucommia")
mutate_show_palette(m.stat_palette, font_size = 25,
             width = 10,
             height = 2,
             title = "",
             fill_lab = "Group",
             legend.position = "bottom",
             legend.key.height = unit(0.5, "cm"),
             legend.key.width = unit(3, "cm"),
             xlab = "", ylab = "", re_order = F)
## ------------------------------------------------------------------------
## nodes_mark
mark_df <- data.table::data.table(.id = "3918",
                                  mark = "ID:3918")
mark_palette <- c(Others = "#D9D9D9", `ID:3918` = "#EFC000")
## Pyranones and derivatives
## Iridoid O-glycosides
nebula_name <- "Iridoid O-glycosides"
vis_via_molconvert_nebulae(nebula_name, .MCn.nebula_index,
                           output="mcnebula_results/trace/tmp/structure")
## ------------------------------------
annotate_child_nebulae(nebula_name,
                       layout = "fr",
                       output="mcnebula_results/trace",
                       ## pie diagrame setting
                       nodes_mark = mark_df,
                       palette = mark_palette,
                       ratio_df = mean.feature_stat,
                       palette_stat = stat_palette,
                       global.node.size = 0.8,
                       theme_args = list(panel.background = element_rect(),
                                         panel.grid = element_line()))
## ------------------------------------------------------------------------
```

```
## show statistic legend for nebula
```

2 File: bio__generate__child__nebula.R

```
## ------------------------------------------------------------------------
## classes
classes <- dplyr::filter(.MCn.nebula_index, .id %in% log.fc_stat$.id)$name %>%
  unique()
f.classes <- dplyr::filter(.data = .MCn.nebula_index, name %in% all_of(classes)) %>%
  .$name %>%
  table() %>%
  .[which(. <= 300)] %>%
  names()
## generate_child_nebulae
tmp_nebula_index <- dplyr::filter(.MCn.nebula_index,
                                  .id %in% log.fc_stat$.id,
                                  name %in% f.classes)
generate_child_nebulae(nebula_index = tmp_nebula_index,
                       output = "mcnebula_results/trace")
```

3 File: bio__visualize__child__nenula.R

```
## ------------------------------------------------------------------------
## for tracing compounds
## set df to mark efs25 nodes
mark_df <- data.table::data.table(.id = paste0(log.fc_stat$.id), mark = log.fc_stat$log2fc)
## set mark_palette
mark_palette <- c("#FED439FF", "#709AE1FF")
names(mark_palette) <- c("10 times FC", "Others")
## visualize_child_nebulae
fill_expression <- "scale_fill_gradient2(low = 'blue', high = 'red', mid = 'white', midpoint = 0)"
visualize_child_nebulae(output = "mcnebula_results/trace",
                        nodes_mark = mark_df,
                        scale_fill_expression = fill_expression,
                        remove_legend_lab = T,
                        legend_fill = T,
                        legend_position = "bottom",
                        width = 15, height = 20)
file.rename("mcnebula_results/trace/child_nebulae.svg", "mcnebula_results/trace/gradient_nebula.svg")
## ------------------------------------------------------------------------
```

```
visualize_child_nebulae(output = "mcnebula_results/trace",
                        width = 15, height = 20)
file.rename("mcnebula_results/trace/child_nebulae.svg", "mcnebula_results/trace/struc_nebula.svg")
## ------------------------------------------------------------------------
## ------------------------------------------------------------------------
## visualize legend for nebula-index
label_palette <- .MCn.palette_label[3:6]
names(label_palette) <- Hmisc::capitalize(c("level 5", "subclass", "class", "superclass"))[4:1]
mutate_show_palette(label_palette, font_size = 25,
             width = 3,
             title = "",
             fill_lab = "Class-Hierarchy",
             xlab = "", ylab = "", re_order = F)
```

4 File: export.extra1_syno.R

```
## -------------------------------------
## get cid and inchikey set
rdata <- paste0("pubchem", "/", "inchikey.rdata")
## extract as list
cid_inchikey <- extract_rdata_list(rdata) %>%
  lapply(function(df){
           if("CID" %in% colnames(df))
             return(df)
  }) %>%
  data.table::rbindlist(idcol = T) %>%
  dplyr::rename(inchikey2D = .id)
## -------------------------------------
## get cid
cid <- cid_inchikey$CID
## create dir
dir.create("syno")
## curl synonyms
pubchem_get_synonyms(cid, dir = "syno", curl_cl = 4)
## ------------------------------------------------------------------------
```

5 File: export.extra2_syno.filter.R

```
## extract data
cid_syno <- extract_rdata_list("syno/cid.rdata") %>%
```

```
  data.table::rbindlist()
## ------------------------------------
## filter the syno, according to character
filter.syno <- cid_syno %>%
  ## na
  dplyr::filter(!is.na(syno)) %>%
  ## specific ID
  dplyr::filter(!stringr::str_detect(syno, "[^[a-z]]{5,}$")) %>%
  ## id filter
  dplyr::filter(!stringr::str_detect(syno, "[0-9]{4,}$")) %>%
  ## alphabet number >= 4
  dplyr::filter(stringr::str_detect(syno, "[a-z]{4,}"))
## ------------------------------------
## show results
filter.syno %>% dplyr::as_tibble() %>% print(n = 20)
```

6 File: export.extra3_read.pdf.R

```
## read pdf as string document
path <- "~/Documents/eucommia/"
lite.bib <- paste0(path, "eucommia.bib")
## ------------------------------------
## use package 'bib2df' to parse .bib file
lite.df <-  bib2df::bib2df(lite.bib)
## do filter
lite.df.file <- lite.df %>%
  ## keep with file
  dplyr::filter(!is.na(FILE)) %>%
  ## absolute path
  dplyr::mutate(FILE = paste0(path, FILE)) %>%
  ## filter according to year
  dplyr::filter(YEAR >= 2000)
## ------------------------------------
## multi threashold read pdf
pdf_list <- pbapply::pblapply(lite.df.file$FILE,
                              function(pdf){
                                check <- try(text <- pdftools::pdf_ocr_text(pdf))
                                if(class(check)[1] == "try-error"){
                                  return("error")
                                }else{
                                  return(text)
```

```
                                }
                              })
## ------------------------------------
```

7 File: export.extra4_df.pre.R

```
## check error number
# check <- rapply(pdf_list,
                # function(vec){
                #   check <- lapply(vec,
                #                   function(vec){
                #                     if(vec == "error")
                #                       return("error")
                #                   })
                #   return(check)
                # }) %>%
  # unlist()
## ------------------------------------------------------------------------
## syno.metadata: filter.syno
## cid_inchikey.metadata: cid_inchikey
## gether syno with inchikey2D
syno.metadata <- merge(filter.syno, cid_inchikey,
                       all.x = T, by.x = "cid", by.y = "CID") %>%
  dplyr::as_tibble()
## ------------------------------------
## pdf.metadata: lite.df.file
## ------------------------------------
## format each pdf
format.lite <- pdf_list %>%
  lapply(paste0, collapse = "") %>%
  lapply(function(str){
           str <- gsub("\n", " ", str)
           return(str)
                       }) %>%
  unlist()
```

8 File: export.extra5.str_dist.R

```
## ------------------------------------
## stringdist::afind
```

```
## test
res.list <- pbapply::pblapply(format.lite,
                              function(lite){
                                res.vec <- stringdist::amatch(syno.metadata$syno,
                                                              lite,
                                                              nthread = 10,
                                                              maxDist = 3)
                                return(res.vec)
                              })
```

9 File: export.step0_manual.select.class.R

```
## ------------------------------------------------------------------------
## extract classyfire data
## do a rough filter
class_df <- extract_rdata_list("classyfire/class.rdata",
                               export.struc_set$inchikey2D) %>%
  data.table::rbindlist(idcol = T) %>%
  dplyr::rename(inchikey2d = .id) %>%
  dplyr::filter(!Level %in% all_of(c("kingdom", "level 7", "level 8", "level 9")),
                !grepl("[0-9]|Organ|Phenylpropanoids and polyketides", Classification))
## ------------------------------------------------------------------------
## the classes keeped
## do further filter
keep <- class_df$Classification %>%
  table() %>%
  data.table::data.table() %>%
  dplyr::filter(N >= 7, N < 100) %>%
  dplyr::rename(class = 1, abund = 2)
## ------------------------------------
## set abundance db
abund.db <- lapply(keep$abund, c)
names(abund.db) <- keep$class
## ------------------------------------
## get parent class
parent.class <- mutate_get_parent_class(keep$class, class_cutoff = 3, this_class = F)
## final auto discard
discard <- mapply(function(c.name, vec){
                    if(is.null(vec)){
                      return()
                    }
```

```
                    vec <- vec[vec %in% names(abund.db)]
                    if(length(vec) == 0)
                      return()
                    vec <- lapply(vec, function(class){abund.db[[class]]}) %>%
                      unlist(use.names = F)
                    ref <- abund.db[[c.name]]
                    check <- vec - ref
                    check <- 0:3 %in% check
                    if(T %in% check){
                      return(c.name)
                    }
                }, names(parent.class), parent.class) %>%
  unlist(use.names = F)
## -------------------------------------
keep2 <- dplyr::filter(keep, !class %in% all_of(discard))
## -------------------------------------
manual_exclude <- c("O-glycosyl compounds",
                    "Oligosaccharides",
                    "Medium-chain hydroxy acids and derivatives",
                    "Monosaccharides",
                    "Benzenediols",
                    "Benzenoids",
                    "Benzoic acids and derivatives",
                    "Aldehydes",
                    "Hydroxy acids and derivatives",
                    "Hydroxy fatty acids",
                    "Keto acids and derivatives",
                    "Ketones",
                    "Hydroxybenzoic acid derivatives",
                    "Benzene and substituted derivatives",
                    "Carbonyl compounds")
## -------------------------------------
keep3 <- dplyr::filter(keep2, !class %in% all_of(manual_exclude))
## ------------------------------------------------------------------------
class_meta <- class_df %>%
  dplyr::filter(Classification %in% keep3$class)
```

10 File: export.step1_class.R

```
## select cols to export
## format table
```

```r
export.all <- export.struc_set %>%
  dplyr::select(.id, name, molecularFormula, tanimotoSimilarity, inchikey2D, smiles) %>%
  dplyr::arrange(inchikey2D, desc(tanimotoSimilarity)) %>%
  dplyr::distinct(inchikey2D, .keep_all = T) %>%
  ## get mz and rt
  merge(mz_rt, by = ".id", all.x = T) %>%
  dplyr::relocate(.id, mz, rt) %>%
  dplyr::as_tibble()
## ------------------------------------
## merge with classyfire Classification
export.class <- export.all %>%
  merge(dplyr::select(class_meta, inchikey2d, Classification),
        by.x = "inchikey2D", by.y = "inchikey2d", all.x = T) %>%
  dplyr::as_tibble()
na.class.id <- dplyr::filter(export.class, is.na(Classification))$.id %>%
  unique()
## ------------------------------------
## get canopus Classification annotation
canopus_anno <- "canopus_summary.tsv" %>%
  read_tsv() %>%
  dplyr::select(1:(ncol(.) - 1), -2, -3) %>%
  dplyr::rename(.id = 1) %>%
  ## get .id
  dplyr::mutate(.id = stringr::str_extract(.id, "[0-9]{1,}$")) %>%
  reshape2::melt(id.vars = ".id", variable.name = "level", value.name = "canopus") %>%
  dplyr::filter(level != "most specific class",
                canopus != "",
                .id %in% na.class.id) %>%
  dplyr::select(.id, canopus) %>%
  dplyr::as_tibble()
## ------------------------------------
## fill na Classification annotation
export.class.cano <- merge(export.class, canopus_anno, all.x = T, by = ".id") %>%
  dplyr::mutate(Classification = ifelse(is.na(Classification), canopus, Classification)) %>%
  dplyr::select(-canopus) %>%
  dplyr::filter(!is.na(Classification)) %>%
  dplyr::as_tibble()
## ------------------------------------------------------------------------
```

11 File: export.step2_name.diff.vari.R

```
## add supplementation data
export.supp <- export.class.cano %>%
  ## get name
  merge(iupac, by.x = "inchikey2D", by.y = "inchikey2d", all.x = T) %>%
  ## get mass difference
  merge(.MCn.formula_set[, c(".id", "massErrorPrecursor(ppm)")], ) %>%
  ## get content variation
  merge(log.fc_stat.ori[, c(".id", "log2fc")], by = ".id", all.x = T) %>%
  dplyr::as_tibble() %>%
  ## ------------------------------------
  ## format data
  dplyr::mutate(name = ifelse(name == "null",
                              ## get better name
                              ifelse(is.na(IUPACName), name, IUPACName),
                              ifelse(nchar(name) < nchar(IUPACName), name, IUPACName)),
                name = ifelse(grepl("^bmse|^ACMC", name), IUPACName, name),
                ## round rt min
                rt = round(rt, 1),
                ## variation
                log2fc = unlist(lapply(log2fc, function(num){
                                    if(abs(num) < 1)
                                      return("-")
                                    sig <- ifelse(num > 0, "↑", "↓")
                                    ch <- paste(rep(sig, floor(abs(num))), collapse = "")
                                    ## log2fc >= 5, omit as ...
                                    if(nchar(ch) >= 5)
                                      ch <- paste(c(rep(sig, 5), "..."), collapse = "")
                                    return(ch)
                                  })))
```

12 File: export.step3_format.R

```
## ----------------------------------------------------------------------
export <- export.supp %>%
  ## exclude useless
  dplyr::select(-IUPACName) %>%
  ## relocate name
  dplyr::relocate(name, .id, mz, `massErrorPrecursor(ppm)`, rt) %>%
  dplyr::relocate(inchikey2D, smiles, .after = last_col()) %>%
```

```
  ## get hierarchy rank
  get_hierarchy.in_df(col = "Classification") %>%
  ## duplicated in class
  dplyr::arrange(desc(hierarchy), inchikey2D) %>%
  dplyr::distinct(inchikey2D, .keep_all = T) %>%
  ## remove hierarchy
  dplyr::select(-hierarchy) %>%
  ## order df
  dplyr::arrange(Classification, name) %>%
  dplyr::rename(id = .id, `precursor m/z` = mz,
                variation = log2fc,
                `RT (min)` = rt, formula = molecularFormula,
                `tanimoto similarity` = tanimotoSimilarity,
                `InChIKey planar` = inchikey2D,
                `mass error (ppm)` = `massErrorPrecursor(ppm)`)
## ------------------------------------
export.dominant <- export %>%
  dplyr::filter(name != "null") %>%
  set_export.no()
## ------------------------------------
export.extra <- export %>%
  dplyr::filter(name == "null")
```

13 File: export.step9_gt.R

```
## ------------------------------------------------------------------------
## export
gt_table <- pretty_table(dplyr::rename(export.dominant[, -ncol(export.dominant)], info = Classification)
  title = "E. ulmoides compounds summary",
  subtitle = "LC-MS in negative ion mode",
  footnote = "Compounds listed in table were identified from Raw-Eucommia or Pro-Eucommia. These compou
  default = F) %>%
## add footnote
## name
tab_footnote(footnote = "The names are synonyms or IUPAC names of these compounds or their stereoisomers
  locations = cells_column_labels(columns = Name)) %>%
## similarity
tab_footnote(footnote = "Tanimoto similarities were obtained via CSI:fingerID for evaluation of compound
  locations = cells_column_labels(columns = `Tanimoto similarity`)) %>%
## InChIKey
tab_footnote(footnote = "The 'InChIKey planar' is the first hash block of InChIKey that represents a mo
```

```
  locations = cells_column_labels(columns = `InChIKey planar`)) %>%
tab_footnote(footnote = "All identified formulae are in adduct of '[M - H]-'.",
  locations = cells_column_labels(columns = `Formula`)) %>%
tab_footnote(footnote = "Ids were generated while MZmine2 processing and were inherited in subsequent M
  locations = cells_column_labels(columns = `Id`)) %>%
tab_footnote(footnote = "The mass error were obtained via SIRIUS while predicting formula of compounds.
  locations = cells_column_labels(columns = `Mass error (ppm)`)) %>%
tab_footnote(footnote = "The variation are difined as: ↑ or ↓ indicates increasing or decreasing of com
  locations = cells_column_labels(columns = `Variation`))
```

14 File: figure.eu_iso.R

```
##  re-draw heartmap for specific id
sp.idset <- c(3380, 2824, 3938, 2664, 2529,
              2227, 3918, 1445, 1107,
              574, 458, 279) %>%
  as.character()
## ------------------------------------
test <- tmp_nebula_index %>%
  dplyr::filter(.id %in% all_of(sp.idset))
```

15 File: get_iupacname.R

```
## via pubchem, fill with name in witch compound
## idenfication table is 'null'
## ------------------------------------
dir.create("iupac_name")
pubchem_curl_inchikey(unique(export.class.cano$inchikey2D),
                      dir = "iupac_name",
                      get = "IUPACName",
                      curl_cl = 8)
## ------------------------------------
## extract data witch has curl down
## format
iupac <- extract_rdata_list("iupac_name/inchikey.rdata") %>%
  data.table::rbindlist(idcol = T, fill = T) %>%
  dplyr::rename(inchikey2d = .id) %>%
  ## filter na
  dplyr::filter(is.na(x), !is.na(IUPACName)) %>%
  dplyr::select(1:3) %>%
```

```
  ## length of character
  dplyr::mutate(n.ch = nchar(IUPACName)) %>%
  dplyr::arrange(inchikey2d, n.ch) %>%
  ## get the unique name
  dplyr::distinct(inchikey2d, .keep_all = T) %>%
  dplyr::select(inchikey2d, IUPACName)
```

16 File: get_real_class.R

```
## stat all identified structure, find
## high confidence score compound
## and collate them for group of classification
## -------------------------------------
## get the real classification via classyfire
export.struc_set <- .MCn.structure_set %>%
  dplyr::filter(tanimotoSimilarity >= 0.5)
## ------------------------------------------------------------------------
## curl pubchem to get possibbly inchikey3d
dir.create("pubchem")
pubchem_curl_inchikey(unique(export.struc_set$inchikey2D), dir = "pubchem",
                      curl_cl = 8)
## -------------------------------------
## classify
dir.create("classyfire")
batch_get_classification(unique(export.struc_set$inchikey2D),
                         dir_pubchem = "pubchem",
                         dir_classyfire = "classyfire",
                         classyfire_cl = 8)
```

17 File: help.export.R

```
## no filter
source("~/outline/mcnebula_cell_validation/export.step0_manual.select.class.R")
source("~/outline/eucommia_neg/export.step1_class.R")
## source(~/outline/eucommia_neg/get_iupacname.R)
source("~/outline/eucommia_neg/export.step2_name.diff.vari.R")
source("~/outline/eucommia_neg/export.step3_format.R")
source("~/outline/eucommia_neg/export.step9_gt.R")
## -------------------------------------
## insert reference
```

```
# export.extra1_syno.R
# export.extra2_syno.filter.R
# export.extra3_read.pdf.R
# export.extra4_df.pre.R
```

18 File: mcnebula_run.R

```
##
load_all("~/MCnebula/R")
load_all("~/extra/R")
initialize_mcnebula(".")
collate_structure()
build_classes_tree_list()
collate_ppcp(min_possess = 30, max_possess_pct = 0.07)
generate_parent_nebula()
generate_child_nebulae()
visualize_parent_nebula()
visualize_child_nebulae(width = 15, height = 20, nodes_size_range = c(2, 4))
```

19 File: stat_ratio.R

```
## stat peak area
mzmine_table <- "mcnebula_results/mzmine_table.tsv" %>%
  read_tsv()
## ------------------------------------
mz_rt <- mzmine_table %>%
  dplyr::select(1:3) %>%
  dplyr::rename(.id = 1, mz = 2, rt = 3) %>%
  dplyr::mutate(.id = as.character(.id))
## ------------------------------------
## reset name
colnames(mzmine_table) <- gsub("\\.mzML Peak area", "", colnames(mzmine_table))
## metadata
meta_feature_stat <- colnames(mzmine_table)[c(-1, -2, -3)] %>%
  data.table::data.table(name = .) %>%
  dplyr::mutate(subgroup = stringr::str_extract(name, ".*(?=[0-9])"),
                subgroup = ifelse(is.na(subgroup), "Blank", subgroup))
## ------------------------------------------------------------------------
## summarise mean of each .id in groups
mean.feature_stat <- mzmine_table %>%
```

```
  dplyr::rename(.id = `row ID`) %>%
  dplyr::select(.id, all_of(meta_feature_stat$name)) %>%
  ## as long table
  reshape2::melt(id.vars = ".id", variable.name = "name", value.name = "value") %>%
  ## get group info
  merge(meta_feature_stat, by = "name", all.x = T) %>%
  data.table::data.table() %>%
  ## as numeric
  dplyr::mutate(value = as.numeric(value)) %>%
  ## group by .id and subgroup
  .[, list(mean = mean(value, na.rm = T)), by = list(.id, subgroup)] %>%
  ## calculate mean
  dplyr::mutate(mean = ifelse(is.nan(mean), 0, mean)) %>%
  ## as wide data
  data.table::dcast(.id ~ subgroup, value.var = "mean") %>%
  dplyr::as_tibble()
## ------------------------------------------------------------------------
## ten times fold change
log.fc_stat.ori <- mean.feature_stat %>%
  dplyr::mutate(`EU-Pro` = `EU-Pro` + 1,
                `EU-Raw` = `EU-Raw` + 1,
                log2fc = log2(`EU-Pro` / `EU-Raw`))
log.fc_stat <- log.fc_stat.ori %>%
  dplyr::filter(abs(log2fc) >= 1)
```

20 File: variation_abundance.R

```
## stat variation relativa abundance
## gather data
var_rel.abund <- list(tmp_nebula_index, .MCn.nebula_index) %>%
  lapply(function(df){
           df <- table(df$name) %>%
             data.table::data.table() %>%
             dplyr::rename(class = V1, abund = N)
           return(df)
               })
## stat rel.adund
var_rel.abund <- merge(var_rel.abund[[1]], var_rel.abund[[2]], by = "class", all.x = T) %>%
  dplyr::mutate(rel.abund = abund.x / abund.y)
## ------------------------------------
generic_horizon_bar(df = dplyr::select(df, class, rel.abund),
```

```
                    xlab = "classification",
                    ylab = "variation relative abundance",
                    save = "mcnebula_results/trace/rank.svg")
## ------------------------------------
# main <- read_svg("mcnebula_results/trace/stru_child_nebulae.svg", as_cairo = F)
# legend <- read_svg("mcnebula_results/trace/rank.svg")
# ## ------------------
# grid_draw_svg.legend(main, legend,
#                      width = 16,
#                      position.main = 0.7,
                     # savename = "mcnebula_results/trace/merge_gradient.svg")
```