R codes of analysis gene

Contents

1 File: analysis_data.1.R

```
## R
## meta
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl")
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ========== Run block ==========
gse <- meta$Accession[1]
set.sig.wd(gse)
meta.df <- decomp_tar2txt()
## -------------------------------------
meta.df <- dplyr::mutate(
  meta.df,
  treatment = stringr::str_extract(
    file, "(?<=D7-).*(?=_RNA-seq)"
    ),
  group = treatment,
  sample = stringr::str_extract(
    file, "(?<=_).*(?=_RNA-seq)"
  )
)
## -------------------------------------
gene.anno.tmp <- data.table::fread(meta.df$file[1]) %>%
  dplyr::select(grep("Symbol|Ensembl|reference sequence", colnames(.))) %>%
  dplyr::mutate(ref = `Locus on reference sequence`,
                chr = stringr::str_extract(ref, "^chr[0-9]{1,}(?=:)"),
                seq.st = stringr::str_extract(ref, "(?<=:)[0-9]{1,}(?=-)"),
                seq.st = as.integer(seq.st),
                seq.end = stringr::str_extract(ref, "(?<=-)[0-9]{1,}$"),
                seq.end = as.integer(seq.end),
                eff.len = abs(seq.st - seq.end)) %>%
  dplyr::relocate(`Ensembl Gene ID`, `eff.len`) %>%
  dplyr::as_tibble()
## ---------------------------------------
dge.list <- edgeR::readDGE(meta.df$file, columns = c(12, 2))
## add group
dge.list <- re.sample.group(dge.list, meta.df)
## add annotation
dge.list <- anno.into.list(dge.list, gene.anno.tmp, "Ensembl Gene ID")
## ---------------------------------------
## fpkm to tpm
dge.list <- fpkm_log2tpm(dge.list)
## ---------------------------------------
group. <- dge.list$samples$group
## design
design <- model.matrix(~ 0 + group.)
## contrast
contr.matrix <- limma::makeContrasts(
  ## treat with CH223191
  treat_ch.vs.contr = group.CH - group.C,
  ## treat with StemRegenin1
  treat_sr.vs.contr = group.SR1 - group.C,
  levels = design
)
## ---------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix,
                        min.count = 0.0001, voom = F)
## ---------------------------------------
## save
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

2 File: analysis_data.10.R

```
## R
## meta
```

```r
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = c("ensembl_exon_id", "go_id"))
## -------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[10]
set.sig.wd(gse)
## -------------------------------------
info <- GEOquery::getGEO(gse)
## -------------------------------------
meta.df.raw <- Biobase::phenoData(info[[1]]) %>%
  Biobase::pData() %>%
  dplyr::as_tibble()
## -------------------------------------
meta.df <- dplyr::select(meta.df.raw, title) %>%
  dplyr::mutate(sample = title,
                group = gsub("_[0-9]{1,}$", "", sample),
                group = gsub("U87_shC_", "", group))
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## contrast
contr.matrix <- limma::makeContrasts(
  treat_ficz.vs.contr = group.FICZ100nM - group.DMSO,
  treat_kyna.vs.contr = group.KynA50uM - group.DMSO,
  levels = design
)
## -------------------------------------
## show expression dataset
# exprs <- Biobase::assayData(info[[1]])$exprs
# print(head(exprs))
## -------------------------------------
# genes <- fit$genes %>%
#   dplyr::as_tibble()
## -------------------------------------
res <- limma_downstream.eset(info[[1]], design, contr.matrix)
## -------------------------------------
## ========== Run block ==========
res <- lapply(res, dplyr::mutate,
              ensembl = stringr::str_extract(gene_assignment, "ENS[A-Z][0-9]*"),
              symbol = stringr::str_extract(gene_assignment,
                "(?<= |^)[A-Z]{1,}[0-9]{0,2}[A-Z]{1,}[0-9]{0,2}[A-Z]{0,}[0-9]{0,3}(?= |$)")) %>%
  lapply(dplyr::relocate, ensembl, symbol)
## -------------------------------------
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

3 File: analysis_data.11.R

```
## R
## meta
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = c("ensembl_exon_id", "go_id"))
## -------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[11]
set.sig.wd(gse)
## -------------------------------------
info <- GEOquery::getGEO(gse)
## -------------------------------------
meta.df.raw <- Biobase::phenoData(info[[1]]) %>%
  Biobase::pData() %>%
  dplyr::as_tibble()
## -------------------------------------
meta.df <- dplyr::select(meta.df.raw, title) %>%
  dplyr::mutate(sample = title,
                group = gsub("_[0-9]{1,}$", "", sample),
                group = gsub("U87_", "", group))
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## contrast
contr.matrix <- limma::makeContrasts(
  treat_i3ca.vs.contr = group.I3CA50uM - group.DMSO,
  levels = design
)
## -------------------------------------
## show expression dataset
exprs <- Biobase::assayData(info[[1]])$exprs
## -------------------------------------
res <- limma_downstream.eset(info[[1]], design, contr.matrix)
## -------------------------------------
```

```
res <- lapply(res, dplyr::mutate,
              ensembl = stringr::str_extract(gene_assignment, "ENS[A-Z][0-9]*"),
              symbol = stringr::str_extract(gene_assignment,
                "(?<= |^)[A-Z]{1,}[0-9]{0,2}[A-Z]{1,}[0-9]{0,2}[A-Z]{0,}[0-9]{0,3}(?= |$)")) %>%
  lapply(dplyr::relocate, ensembl, symbol)
## -------------------------------------
## ========== Run block ==========
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

4 File: analysis_data.12.R

```
## R
## meta
## ---------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = c("ensembl_exon_id", "go_id"))
## -------------------------------------
## ---------------------------------------------------------------------
gse <- meta$Accession[12]
set.sig.wd(gse)
## -------------------------------------
info <- GEOquery::getGEO(gse)
## -------------------------------------
meta.df.raw <- Biobase::phenoData(info[[1]]) %>%
  Biobase::pData() %>%
  dplyr::as_tibble()
## -------------------------------------
meta.df <- dplyr::select(meta.df.raw, title) %>%
  dplyr::mutate(sample = title,
                group = gsub("_[0-9]{1,}$", "", sample),
                group = gsub("^.*U87_", "", group))
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## -------------------------------------
## contrast
contr.matrix <- limma::makeContrasts(
  treat_hpp.vs.contr = group.HPP40uM - group.DMSO,
  treat_i3p.vs.contr = group.I3P40uM - group.DMSO,
  treat_pp.vs.contr = group.PP40uM - group.DMSO,
  levels = design
)
## -------------------------------------
## show expression dataset
# exprs <- Biobase::assayData(info[[1]])$exprs
## -------------------------------------
res <- limma_downstream.eset(info[[1]], design, contr.matrix)
## -------------------------------------
res <- lapply(res, dplyr::mutate,
              ensembl = stringr::str_extract(gene_assignment, "ENS[A-Z][0-9]*"),
              symbol = stringr::str_extract(gene_assignment,
                "(?<= |^)[A-Z]{1,}[0-9]{0,2}[A-Z]{1,}[0-9]{0,2}[A-Z]{0,}[0-9]{0,3}(?= |$)")) %>%
  lapply(dplyr::relocate, ensembl, symbol)
## -------------------------------------
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

5 File: analysis_data.13.R

```
## R
## meta
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = c("ensembl_exon_id", "go_id"))
## -------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[13]
set.sig.wd(gse)
## -------------------------------------
info <- GEOquery::getGEO(gse)
## -------------------------------------
meta.df.raw <- Biobase::phenoData(info[[1]]) %>%
  Biobase::pData() %>%
  dplyr::as_tibble()
## -------------------------------------
meta.df <- dplyr::select(meta.df.raw, title) %>%
  dplyr::mutate(sample = title,
                group = gsub("_[0-9]{1,}$", "", sample),
                group = gsub("^.*U87_", "", group))
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## -------------------------------------
```

```r
## contrast
contr.matrix <- limma::makeContrasts(
  ## treat with L-trp
#   il4i1_ahrKO.vs.ahrKO_1 = group.IL4I1_shAHR1 - group.C_shAHR1,
#   il4i1_ahrKO.vs.ahrKO_2 = group.IL4I1_shAHR2 - group.C_shAHR2,
  ahrKO_1.vs.control = group.C_shAHR1 - group.C_shC,
  ahrKO_2.vs.control = group.C_shAHR2 - group.C_shC,
  levels = design
)
## -------------------------------------
## show expression dataset
# exprs <- Biobase::assayData(info[[1]])$exprs
## -------------------------------------
res <- limma_downstream.eset(info[[1]], design, contr.matrix)
## -------------------------------------
res <- lapply(res, dplyr::mutate,
              ensembl = stringr::str_extract(gene_assignment, "ENS[A-Z][0-9]*"),
              symbol = stringr::str_extract(gene_assignment,
                "(?<= |^)[A-Z]{1,}[0-9]{0,2}[A-Z]{1,}[0-9]{0,2}[A-Z]{0,}[0-9]{0,3}(?= |$)")) %>%
  lapply(dplyr::relocate, ensembl, symbol)
## -------------------------------------
## ========== Run block ==========
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

6 File: analysis_data.16.R

```r
## R
## meta
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = c("ensembl_exon_id", "go_id"))
## -------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[16]
set.sig.wd(gse)
## -------------------------------------
info <- GEOquery::getGEO(gse)
## -------------------------------------
meta.df.raw <- Biobase::phenoData(info[[1]]) %>%
  Biobase::pData() %>%
  dplyr::as_tibble()
```

```r
## -------------------------------------
meta.df <- dplyr::select(meta.df.raw, title) %>%
  dplyr::mutate(sample = title,
                group = gsub(" [0-9]{1,}$", "", sample),
                group = gsub("-", "_", group))
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## -------------------------------------
## contrast
contr.matrix <- limma::makeContrasts(
  treat.vs.control = group.Co_culture - group.Single_culture,
  levels = design
)
## -------------------------------------
## show expression dataset
# exprs <- Biobase::assayData(info[[1]])$exprs
## -------------------------------------
res <- limma_downstream.eset(info[[1]], design, contr.matrix)
## -------------------------------------
res <- lapply(res, dplyr::mutate,
              ensembl = stringr::str_extract(SPOT_ID.1, "ENS[A-Z][0-9]*"),
              symbol = stringr::str_extract(SPOT_ID.1,
                "(?<=\\()[A-Z]{1,}[0-9]{0,2}[A-Z]{1,}[0-9]{0,2}[A-Z]{0,}[0-9]{0,3}(?=\\))")) %>%
  lapply(dplyr::relocate, ensembl, symbol)
## -------------------------------------
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

7 File: analysis_data.17.R

```r
## R
## meta
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = c("ensembl_exon_id", "go_id"))
## -------------------------------------
check <- 0
n <- 0
while(check == 0){
  n <- n + 1
  check <- try(gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl",
                                              ex.attr = c("go_id", "refseq_mrna")),
               silent = T)
  if(class(check)[1] == "try-error"){
    print(check)
    check <- 0
  }else{
    check <- 1
  }
  cat("##", "Try...", n, "\n")
}
## ----------------------------------------------------------------------
gse <- meta$Accession[17]
set.sig.wd(gse)
## -------------------------------------
meta.df <- decomp_tar2txt()
## -------------------------------------
meta.df <- dplyr::mutate(meta.df, sample = gsub("^GSM[^_]*_", "", file),
                         sample = gsub("\\.txt$", "", sample),
                         group = gsub("_60_S.*_pool.*$", "", sample))
## -------------------------------------
## format
raw <- lapply(meta.df$file, data.table::fread) %>%
  lapply(dplyr::distinct, tracking_id, .keep_all = T) %>%
  lapply(dplyr::rename, fpkm = 6) %>%
  lapply(dplyr::mutate, fpkm = as.numeric(fpkm))
## filter NA
nas <- lapply(raw, dplyr::filter, is.na(fpkm)) %>%
  lapply(`[[`, "tracking_id") %>%
  unlist(use.names = F) %>%
  unique()
raw <- lapply(raw, dplyr::filter, !tracking_id %in% all_of(nas))
## -------------------------------------
gene.anno.tmp <- dplyr::select(raw[[1]], 1:2)
## write into disk
mapply(write_tsv, raw, meta.df$file)
## -------------------------------------
dge.list <- edgeR::readDGE(meta.df$file, columns = c(1, 6))
## add group
dge.list <- re.sample.group(dge.list, meta.df)
## -------------------------------------
## add annotation
```

```
dge.list <- anno.into.list(dge.list, gene.anno.tmp, "tracking_id")
## -------------------------------------
dge.list <- fpkm_log2tpm(dge.list)
## -------------------------------------
group. <- dge.list$samples$group
## design
design <- model.matrix(~ 0 + group.)
## contrast
## -------------------------------------
contr.matrix <- limma::makeContrasts(
  treat_bap.vs.contr = group.MCF10AT1_BA_P - group.MCF10AT1_NT,
  treat_bpa.vs.contr = group.MCF10AT1_BPA - group.MCF10AT1_NT,
  treat_overlay.vs.contr = group.MCF10AT1_BPAplus_Ba_P - group.MCF10AT1_NT,
  levels = design
)
## -------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix,
                        min.count = 0.0001, voom = F)
## -------------------------------------
## ========== Run block ==========
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

8 File: analysis_data.18.R

```
## R
## meta
## ----------------------------------------------------------------------
gse <- meta$Accession[18]
set.sig.wd(gse)
## -------------------------------------
list.files(pattern = "\\.gz$") %>%
  R.utils::gunzip()
## -------------------------------------
## ========== Run block ==========
raw <- data.table::fread("GSE130234_processed_data.txt") %>%
  dplyr::as_tibble()
## -------------------------------------
mapply(2:ncol(raw), colnames(raw)[2:ncol(raw)],
       FUN = function(col, name){
         df <- raw[, c(1, col)]
         write_tsv(df, paste0(name, "_counts.tsv"))
```

```r
       })
## -------------------------------------
meta.df <- data.table::data.table(
  file = list.files(pattern = "_counts.tsv$")
) %>%
  dplyr::mutate(sample = gsub("_tagcount_counts.tsv", "", file),
                sample = gsub("-", "_", sample),
                group = gsub("_rep.", "", sample))
## -------------------------------------
## annotation
gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl",
                               attr = c("ensembl_gene_id", "hgnc_symbol", "refseq_mrna"))
## -------------------------------------
dge.list <- edgeR::readDGE(meta.df$file, columns = c(1, 2))
## add group
dge.list <- re.sample.group(dge.list, meta.df)
## add annotation
dge.list <- anno.into.list(dge.list, gene.anno, "refseq_mrna")
## -------------------------------------
# keeps <- !duplicated(dge.list$genes$ensembl_gene_id) | grepl("^NM_", dge.list$genes$refseq_mrna)
# ## filter...
# dge.list <- edgeR::`[.DGEList`(dge.list, keeps, , keep.lib.sizes = F)
## -------------------------------------
group. <- dge.list$samples$group
## design
design <- model.matrix(~ 0 + group.)
## contrast
contr.matrix <- limma::makeContrasts(
  treat_sga315.vs.contr = group.LOV_SGA315 - group.LOV,
  treat_sga360.vs.contr = group.LOV_SGA360 - group.LOV,
  levels = design
)
## -------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix)
## -------------------------------------
## remove mRNA sequences of non-coding proteins
res <- lapply(res, dplyr::filter, !grepl("^NR_", refseq_mrna)) %>%
  lapply(dplyr::relocate, ensembl_gene_id, hgnc_symbol) %>%
  ## remove duplicated genes
  lapply(dplyr::distinct, ensembl_gene_id, .keep_all = T)
## -------------------------------------
## distribution
# exprs <- limma_downstream(dge.list, group., design, contr.matrix, get_normed.exprs = T)
## -------------------------------------
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

9 File: analysis_data.19.R

```r
## R
## meta
## ----------------------------------------------------------------------
gse <- meta$Accession[19]
set.sig.wd(gse)
## -------------------------------------
info <- GEOquery::getGEO(gse)
## -------------------------------------
meta.df.raw <- Biobase::phenoData(info[[1]]) %>%
  Biobase::pData() %>%
  dplyr::as_tibble()
## -------------------------------------
meta.df <- dplyr::select(meta.df.raw, title) %>%
  dplyr::mutate(sample = title,
                group = gsub("^HepG2 ", "", sample),
                group = gsub(" ", "_", group))
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## -------------------------------------
## contrast
contr.matrix <- limma::makeContrasts(
  treat_mc3_1h.vs.control = group.MC3_1_h - group.NT,
  treat_mc3_24h.vs.control = group.MC3_24_h - group.NT,
  levels = design
)
## -------------------------------------
## show expression dataset
# exprs <- Biobase::assayData(info[[1]])$exprs
## -------------------------------------
# res <- limma_downstream.eset(info[[1]], design, contr.matrix)
```

10 File: analysis_data.2.R

```
## R
## meta
## -----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl")
## -----------------------------------------------------------------------
## -----------------------------------------------------------------------
## -----------------------------------------------------------------------
gse <- meta$Accession[2]
set.sig.wd(gse)
meta.df <- decomp_tar2txt()
## -------------------------------------
meta.df <- dplyr::filter(meta.df, !grepl("peaks.txt", file)) %>%
  dplyr::mutate(
    treatment = stringr::str_extract(
      file, "(?<=D7-).*(?=_RNA-seq)"
      ),
    group = treatment,
    sample = stringr::str_extract(
      file, "(?<=_).*(?=_RNA-seq)"
    )
  )
## -------------------------------------
gene.anno.tmp <- data.table::fread(meta.df$file[1]) %>%
  dplyr::select(grep("Symbol|Ensembl|reference sequence", colnames(.))) %>%
  dplyr::mutate(ref = `Locus on reference sequence`,
                chr = stringr::str_extract(ref, "^chr[0-9]{1,}(?=:)"),
                seq.st = stringr::str_extract(ref, "(?<=:)[0-9]{1,}(?=-)"),
                seq.st = as.integer(seq.st),
                seq.end = stringr::str_extract(ref, "(?<=-)[0-9]{1,}$"),
                seq.end = as.integer(seq.end),
                eff.len = abs(seq.st - seq.end)) %>%
  dplyr::relocate(`Ensembl Gene ID`, `eff.len`) %>%
  dplyr::as_tibble()
## -------------------------------------
dge.list <- edgeR::readDGE(meta.df$file, columns = c(12, 2))
## add group
dge.list <- re.sample.group(dge.list, meta.df)
## add annotation
dge.list <- anno.into.list(dge.list, gene.anno.tmp, "Ensembl Gene ID")
```

```
## fpkm to tpm
dge.list <- fpkm_log2tpm(dge.list)
## -------------------------------------
## group
group. <- dge.list$samples$group
## design
design <- model.matrix(~ 0 + group.)
## contrast
contr.matrix <- limma::makeContrasts(
  ## treat with CH223191
  treat_ch.vs.contr = group.CH - group.C,
  ## treat with StemRegenin1
  treat_sr.vs.contr = group.SR1 - group.C,
  levels = design
)
## -------------------------------------
## ========== Run block ==========
res <- limma_downstream(dge.list, group., design, contr.matrix,
                        min.count = 0.0001, voom = F)
## -------------------------------------
## save
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

11 File: analysis_data.20.R

```
## R
## meta
## ----------------------------------------------------------------------
gse <- meta$Accession[20]
set.sig.wd(gse)
## -------------------------------------
list.files(pattern = "\\.gz$") %>%
  R.utils::gunzip()
## -------------------------------------
raw <- data.table::fread("GSE104869_tablecounts.txt") %>%
  dplyr::as_tibble()
## -------------------------------------
## remove the duplicated geneid
raw <- dplyr::mutate(raw, Geneid = gsub("\\.[0-9]$", "", Geneid)) %>%
  dplyr::distinct(Geneid, .keep_all = T)
## -------------------------------------
```

```r
mapply(2:ncol(raw), colnames(raw)[2:ncol(raw)],
       FUN = function(col, name){
         df <- raw[, c(1, col)]
         write_tsv(df, paste0(name, "_counts.tsv"))
       })
## -------------------------------------
meta.df <- data.table::data.table(
  file = list.files(pattern = "_counts.tsv$")
) %>%
  dplyr::mutate(sample = gsub("_counts.tsv", "", file),
                sample = gsub(" ", "_", sample),
                sample = gsub("\\+", "_plus_", sample),
                group = gsub("[0-9]$", "", sample))
## -------------------------------------
gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl",
                               attr = c("ensembl_gene_id", "hgnc_symbol", "refseq_mrna"))
## -------------------------------------
dge.list <- edgeR::readDGE(meta.df$file, columns = c(1, 2))
## add group
dge.list <- re.sample.group(dge.list, meta.df)
## add annotation
dge.list <- anno.into.list(dge.list, gene.anno, "hgnc_symbol")
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## -------------------------------------
## contrast
contr.matrix <- limma::makeContrasts(
  treat_R.vs.control = group.501Mel_R - group.501Mel_Ctrl,
  treat_T.vs.control = group.501Mel_T - group.501Mel_Ctrl,
  levels = design
)
## -------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix)
## -------------------------------------
res <- lapply(res, dplyr::relocate, ensembl_gene_id, hgnc_symbol) %>%
  lapply(dplyr::filter, !is.na(ensembl_gene_id))
## -------------------------------------
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

12 File: analysis_data.21.R

```
## R
## meta
## ----------------------------------------------------------------------
gse <- meta$Accession[21]
set.sig.wd(gse)
## -------------------------------------
meta.df <- decomp_tar2txt()
## -------------------------------------
meta.df <- dplyr::mutate(
  meta.df,
  sample = stringr::str_extract(file, "(?<=_).*(?=\\.count)"),
  sample = gsub("-", "_", sample),
  group = gsub("_[0-9]$", "", sample)
)
## -------------------------------------
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl",
#                                 attr = c("ensembl_gene_id", "hgnc_symbol", "refseq_mrna"))
## -------------------------------------
dge.list <- edgeR::readDGE(meta.df$file, columns = c(1, 2))
## add group
dge.list <- re.sample.group(dge.list, meta.df)
## add annotation
dge.list <- anno.into.list(dge.list, gene.anno, "hgnc_symbol")
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## -------------------------------------
## contrast
contr.matrix <- limma::makeContrasts(
  treat_d1.vs.control = group.TCDD_d1 - group.DMSO_d1,
  treat_d2.vs.control = group.TCDD_d2 - group.DMSO_d2,
  levels = design
)
## -------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix)
## -------------------------------------
res <- lapply(res, dplyr::relocate, ensembl_gene_id, hgnc_symbol) %>%
  lapply(dplyr::filter, !is.na(ensembl_gene_id))
## -------------------------------------
## ========== Run block ==========
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

13 File: analysis_data.22.R

```
## R
## meta
## -----------------------------------------------------------------------
## ========== Run block ==========
gse <- meta$Accession[22]
set.sig.wd(gse)
## --------------------------------------
```

14 File: analysis_data.23.R

```
## R
## meta
## -----------------------------------------------------------------------
gse <- meta$Accession[23]
set.sig.wd(gse)
## --------------------------------------
list.files(pattern = "\\.gz") %>%
  lapply(R.utils::gunzip)
## --------------------------------------
raw <- data.table::fread("GSE116637_counts.txt") %>%
  dplyr::mutate(Geneid = gsub("-[0-9]{1,}$", "", Geneid)) %>%
  dplyr::distinct(Geneid, .keep_all = T) %>%
  dplyr::as_tibble()
## --------------------------------------
lapply(7:ncol(raw), function(col){
         df <- raw[, c(1:6, col)]
         file <- colnames(raw)[col]
         write_tsv(df, paste0(file, "_counts.tsv"))
})
## --------------------------------------
meta.df <- data.table::data.table(
  file = list.files(pattern = "_counts.tsv$")
) %>%
  dplyr::mutate(sample = gsub("_counts.tsv", "", file),
                group = gsub("_[0-9]$", "", sample),
                cell = stringr::str_extract(group, "^[^_]{1,}"),
                agonist = stringr::str_extract(group, "[^_]{1,}$"))
cell.type <- meta.df$cell %>%
  unique()
agonist.type <- meta.df$agonist %>%
  unique()
## -------------------------------------
gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl",
                               attr = c("ensembl_gene_id", "hgnc_symbol", "refseq_mrna"))
## -------------------------------------
dge.list <- edgeR::readDGE(meta.df$file, columns = c(1, 7))
## add group
dge.list <- re.sample.group(dge.list, meta.df)
## add annotation
dge.list <- anno.into.list(dge.list, gene.anno, "hgnc_symbol")
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## -------------------------------------
contr <- data.table::data.table(
  treat = c("3MC", "GNF"),
  control = "DMSO"
)
contr <- lapply(cell.type, function(cell){
                  dplyr::mutate(contr,
                    .treat = treat,
                    treat = paste0("group.", cell, "_", treat),
                    control = paste0("group.", cell, "_", control),
                    contr = paste0(treat, " - ", control),
                    name = paste0("treat_", cell, "_", .treat, ".vs.control"))
})
contr <- data.table::rbindlist(contr)
args <- lapply(contr$contr, function(text){
                 parse(text = text)
})
names(args) <- contr$name
args$levels <- design
## contrast matrix
contr.matrix <- do.call(limma::makeContrasts, args)
## -------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix)
## -------------------------------------
## ========== Run block ==========
res <- lapply(res, dplyr::relocate, ensembl_gene_id, hgnc_symbol) %>%
  lapply(dplyr::filter, !is.na(ensembl_gene_id))
## save
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

15 File: analysis_data.24.R

```
## R
## meta
## ---------------------------------------------------------------------
gse <- meta$Accession[24]
set.sig.wd(gse)
## -------------------------------------
```

16 File: analysis_data.25.R

```
## R
## meta
## ---------------------------------------------------------------------
## ========== Run block ==========
gse <- meta$Accession[25]
set.sig.wd(gse)
## -------------------------------------
```

17 File: analysis_data.26.R

```
## R
## meta
## ---------------------------------------------------------------------
gse <- meta$Accession[26]
set.sig.wd(gse)
## -------------------------------------
meta.df <- decomp_tar2txt()
## -------------------------------------
raw <- lapply(meta.df$file, data.table::fread)
## -------------------------------------
names(raw) <- meta.df$file
raw <- lapply(raw, dplyr::rename, symbol = 4, counts = 6) %>%
  lapply(dplyr::relocate, symbol, counts) %>%
  lapply(dplyr::distinct, symbol, .keep_all = T)
## save as tibble
mapply(raw, names(raw), FUN = function(df, file){
         write_tsv(df, file)
})
## -------------------------------------
## ========== Run block ==========
meta.df <- dplyr::mutate(meta.df,
                         sample = gsub("^GSM[0-9]{1,}_|_RPKM.txt", "", file),
                         group = gsub("^[0-9]{1,}-", "", sample),
                         group = gsub("-", "_", group),
                         .group = group,
                         block = stringr::str_extract(.group, "^[^_]{1,}"),
                         group = gsub("^[^_]{1,}", "AML", .group),
                         cell = stringr::str_extract(group, "^[^_]*(?=_)"),
                         agonist = stringr::str_extract(group, "(?<=_).*$"))
cell.type <- meta.df$cell %>%
  unique()
agonist.type <- meta.df$agonist %>%
  unique()
## -------------------------------------
gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl",
                               attr = c("ensembl_gene_id", "hgnc_symbol", "refseq_mrna"))
## -------------------------------------
dge.list <- edgeR::readDGE(meta.df$file, columns = c(1, 2))
## add group
dge.list <- re.sample.group(dge.list, meta.df)
## add annotation
dge.list <- anno.into.list(dge.list, gene.anno, "hgnc_symbol")
## -------------------------------------
group. <- meta.df$group
design <- model.matrix(~ 0 + group.)
## -------------------------------------
contr <- data.table::data.table(
  treat = agonist.type[3:length(agonist.type)],
  control = "DMSO"
)
contr <- lapply(cell.type, function(cell){
                  dplyr::mutate(contr,
                    .treat = treat,
                    treat = paste0("group.", cell, "_", treat),
                    control = paste0("group.", cell, "_", control),
                    contr = paste0(treat, " - ", control),
                    name = paste0("treat_", cell, "_", .treat, ".vs.control"))
})
contr <- data.table::rbindlist(contr) %>%
  dplyr::filter(treat %in% colnames(design))
args <- lapply(contr$contr, function(text){
                 parse(text = text)
})
names(args) <- contr$name
args$levels <- design
## -------------------------------------
## contrast matrix
contr.matrix <- do.call(limma::makeContrasts, args)
## -------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix, block = meta.df$block)
## -------------------------------------
res <- lapply(res, dplyr::relocate, ensembl_gene_id, hgnc_symbol) %>%
  lapply(dplyr::filter, !is.na(ensembl_gene_id))
## save
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

18 File: analysis_data.27.R

```
## R
## meta
## ----------------------------------------------------------------------
## ========== Run block ==========
gse <- meta$Accession[27]
set.sig.wd(gse)
## -------------------------------------
```

19 File: analysis_data.28.R

```
## R
## meta
## ----------------------------------------------------------------------
## ========== Run block ==========
gse <- meta$Accession[28]
set.sig.wd(gse)
```

```
## -------------------------------------
```

20 File: analysis_data.3.R

```
## R
## meta
## ----------------------------------------------------------------------
## annotation
gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl")
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[3]
set.sig.wd(gse)
# meta.df <- decomp_tar2txt()
## get infoma...
info <- GEOquery::getGEO(gse)
## -------------------------------------
## download data respectively
get_gsm.data(info)
## -------------------------------------
# control (scramble) siRNA + DMSO (GSM5579231, GSM5579232) vs
# control (scramble) siRNA + TCDD (GSM5579233)
meta.df <- data.table::data.table(
  file.xlsx = list.files(pattern = "^GSM557923[1-3]{1}_.*\\.xlsx")
) %>%
  dplyr::mutate(group = rep(c("control", "treatment"), c(2, 1)),
                file = gsub("\\.xlsx$", ".txt", file.xlsx),
                sample = c("control_1", "control_2", "treat"))
## -------------------------------------
## format data
mapply(meta.df$file.xlsx, meta.df$file,
       FUN = function(xlsx, txt){
         df <- readxl::read_xlsx(xlsx) %>%
           dplyr::relocate(ENSEMBL, `Total exon reads`)
         write_tsv(df, txt)
       })
## -------------------------------------
gene.anno.tmp <- data.table::fread(meta.df$file[1]) %>%
  dplyr::select(ENSEMBL, Name, Chromosome, `Exon length`)
## -------------------------------------
```

```r
dge.list <- edgeR::readDGE(meta.df$file)
## group
dge.list <- re.sample.group(dge.list, meta.df)
## annotation
dge.list <- anno.into.list(dge.list, gene.anno.tmp, "ENSEMBL")
## -------------------------------------
group. <- dge.list$samples$group
## design
design <- model.matrix(~ 0 + group.)
## contrast
contr.matrix <- limma::makeContrasts(
  treat.vs.contr = group.treatment - group.control,
  levels = design
)
## -------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix)
## save
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

21 File: analysis_data.4.R

```r
## R
## meta
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl")
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[4]
set.sig.wd(gse)
## unzip
list.files(pattern = "\\.gz") %>%
  lapply(R.utils::gunzip)
## -------------------------------------
## ========== Run block ==========
df <- readxl::read_excel("GSE183606_differentially_expressed_genes.xlsx") %>%
  ## rutaecarpin vs contral
  dplyr::filter(abs(`Log2FC(Rutaecar/Control)`) > 0.3, Padjust < 0.05)
write_tsv(df, "treat.vs.contr_results.tsv")
```

22 File: analysis_data.5.R

```
## R
## meta
## ----------------------------------------------------------------------
## annotation
# attr <- list.attr.biomart()
# check <- 0
# n <- 0
# while(check == 0){
#   n <- n + 1
#   check <- try(gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = "go_id"),
#                silent = T)
#   if(class(check)[1] == "try-error"){
#     print(check)
#     check <- 0
#   }else{
#     check <- 1
#   }
#   cat("##", "Try...", n, "\n")
# }
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = "go_id")
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[5]
set.sig.wd(gse)
## unzip
list.files(pattern = "\\.gz") %>%
  lapply(R.utils::gunzip)
## -------------------------------------
## ========== Run block ==========
res <- data.table::fread("GSE188657_processed_data_files.txt") %>%
  dplyr::filter(abs(log2FoldChange) > 0.3, qValue < 0.05) %>%
  dplyr::as_tibble()
# treat.vs.contr:: AhR antagonist (StemRegenin 1) vs DMSO
```

23 File: analysis_data.6.R

```
## R
## meta
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = "go_id")
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[6]
set.sig.wd(gse)
## -------------------------------------
info <- GEOquery::getGEO(gse)
## -------------------------------------
info <- info[1]
get_gsm.data(info)
## -------------------------------------
list.files(pattern = ".tsv.gz$", recursive = T, full.names = T) %>%
  sapply(function(path){
           system(paste("mv", path, "-t ."))
})
## -------------------------------------
list.files(pattern = ".tsv.gz$", recursive = T, full.names = T) %>%
  sapply(R.utils::gunzip)
## -------------------------------------
## metadata
meta.df <- data.table::fread("metadata.csv", header = F) %>%
  dplyr::rename(sample = 1, anno = 2) %>%
  dplyr::mutate(group = ifelse(grepl("Untreated", anno), "control", "treatment"),
                time = stringr::str_extract(anno, "(?<=_)[0-9]{1,}(?=hr_)"),
                group = paste0(group, "_", time),
                file = paste0(sample, ".tsv"))
## -------------------------------------
## separate annotation
gsm.file <- list.files(pattern = "abundance.tsv$") %>%
  sapply(function(file){
           df <- data.table::fread(file) %>%
             dplyr::mutate(ensembl.v = stringr::str_extract(target_id,
                                                            "(?<=\\|)ENSG[0-9]{1,}[^\\|]{1,}(?=\\|)"),
                           ensembl = stringr::str_extract(ensembl.v, "^ENSG[0-9]{1,}")) %>%
             dplyr::relocate(ensembl, tpm) %>%
             dplyr::distinct(ensembl, .keep_all = T)
           file <- stringr::str_extract(file, "^GSM[0-9]{1,}(?=_)")
           write_tsv(df, paste0(file, ".tsv"))
```

```r
           return(file)
})
## -------------------------------------
gene.anno.tmp <- data.table::fread(meta.df$file[1]) %>%
  dplyr::select(ensembl, ensembl.v, eff_length)
## -------------------------------------
dge.list <- edgeR::readDGE(meta.df$file)
## add group
dge.list <- re.sample.group(dge.list, meta.df)
## add annotation
dge.list <- anno.into.list(dge.list, gene.anno, "ensembl_gene_id")
## log2 tpm
dge.list$counts <- apply(dge.list$counts, 2,
                         function(vec){
                           log2(vec + 1)
                         })
## -------------------------------------
group. <- dge.list$samples$group
## design
design <- model.matrix(~ 0 + group.)
## -------------------------------------
## contrast
contr.matrix <- limma::makeContrasts(
  treat.vs.contr_6 = group.treatment_6 - group.control_6,
  treat.vs.contr_18 = group.treatment_18 - group.control_18,
  treat.vs.contr_72 = group.treatment_72 - group.control_72,
  levels = design
)
## -------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix,
                        min.count = 0.0001, voom = F)
## -------------------------------------
## ========== Run block ==========
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

24 File: analysis_data.7.R

```r
## R
## meta
## ----------------------------------------------------------------------
## annotation
```

```
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = "go_id")
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[7]
set.sig.wd(gse)
## -------------------------------------
list.files(pattern = "txt\\.gz$", recursive = T, full.names = T) %>%
  sapply(R.utils::gunzip)
## -------------------------------------
raw.res <- list.files(pattern = "processed") %>%
  lapply(data.table::fread) %>%
  lapply(function(df){
           df[1:2, ]
})
## -------------------------------------
raw.res <- lapply(raw.res, function(df){
                    mutate.df <- data.table::data.table(
                      ncol = 1:ncol(df),
                      contrast = unlist(df[1, ], use.names = F),
                      type = unlist(df[2, ], use.names = F)
                    )
                    return(mutate.df)
})
## -------------------------------------
form.res <- lapply(raw.res, function(df){
                     dplyr::filter(df, contrast == "" | grepl(" vs ", contrast))
}) %>%
  lapply(by_group_as_list, colnames = "contrast")
## the annotation col
ex.anno.cal <- form.res[[1]][[1]]
## contrast col
form.res <- lapply(form.res, function(lst){
                     lst[[1]] <- NULL
                     lst
})
## -------------------------------------
raw.res <- list.files(pattern = "processed") %>%
  lapply(data.table::fread, skip = 2, header = F)
## -------------------------------------
res <- mapply(form.res, raw.res,
              SIMPLIFY = F,
              FUN = function(form, raw){
                ## convert data.table to tibble
                raw <- dplyr::as_tibble(raw)
                lst <- lapply(form, raw = raw,
                              FUN = function(entry, raw){
                                col <- c(1:4, entry$ncol)
                                ## colnames of data.frame
                                col.name <- c(ex.anno.cal$type, "log2FC", "p-value", "q-value")
                                ## extract column
                                df <- raw[, col]
                                colnames(df) <- col.name
                                ## filter data
                                df <- dplyr::filter(df, abs(log2FC) > 0.3, `q-value` < 0.05)
                                return(df)
                    })
              })
## ------------------------------------
res <- unlist(res, recursive = F)
## ------------------------------------
contrast.entry <- names(res)
contrast <- contrast.entry[c(3, 8, 9, 10)]
res <- res[names(res) %in% contrast]
## ------------------------------------
## ========== Run block ==========
mapply(res, names(res),
       FUN = function(df, names){
         write_tsv(df, paste0(names, "_results.tsv"))
              })
```

25 File: analysis_data.8.R

```
## R
## meta
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = "go_id")
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[8]
```

```r
set.sig.wd(gse)
## -------------------------------------
res.raw <- list.files(pattern = "Results.xlsx") %>%
  readxl::read_xlsx()
## -------------------------------------
res.raw <- dplyr::select(res.raw, 1:4, contains(c("Log2FC", "pAdj")))
## -------------------------------------
res <- reshape2::melt(res.raw, id.vars = colnames(res.raw)[1:4],
                      variable.name = "type",
                      value.name = "value") %>%
  dplyr::as_tibble() %>%
  dplyr::mutate(contrast = gsub("_Log2FC|_pAdj", "", type),
                stat = stringr::str_extract(type, "(?<=_)[^_]{1,}$")) %>%
  dplyr::select(-type) %>%
  by_group_as_list("contrast")
## -------------------------------------
res <- lapply(res, tidyr::spread, key = stat, value = value)
## -------------------------------------
res <- lapply(res, dplyr::filter, abs(Log2FC) > 0.3, pAdj < 0.05)
## -------------------------------------
res <- lapply(res, dplyr::mutate, Ensembl = gsub("\\.[0-9]{1,}$", "", GeneID)) %>%
  lapply(dplyr::relocate, Ensembl, ID)
## -------------------------------------
contrast <- names(res)[c(1, 2, 3, 4)]
res <- res[names(res) %in% contrast]
## -------------------------------------
## ========== Run block ==========
mapply(res, names(res),
       FUN = function(df, names){
         write_tsv(df, paste0(names, "_results.tsv"))
              })
```

26 File: analysis_data.9.R

```r
## R
## meta
## ----------------------------------------------------------------------
## annotation
# gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl", ex.attr = "go_id")
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
## ----------------------------------------------------------------------
gse <- meta$Accession[9]
set.sig.wd(gse)
## -------------------------------------
list.files(pattern = "\\.gz$") %>%
  R.utils::gunzip()
## -------------------------------------
raw <- list.files(pattern = "\\.tab$") %>%
  data.table::fread() %>%
  dplyr::as_tibble()
## -------------------------------------
lapply(2:ncol(raw), function(ncol){
         file <- paste0(colnames(raw)[ncol], ".tsv")
         write_tsv(raw[, c(1, ncol)], file)
})
## -------------------------------------
meta.df <- data.table::data.table(
  file = list.files(pattern = "[0-9]\\.tsv$")
) %>%
dplyr::mutate(
  sample = gsub("\\.tsv$", "", file),
  group = gsub("_[0-9]$", "", sample),
  group = gsub("-", "_", group)
)
## -------------------------------------
dge.list <- edgeR::readDGE(meta.df$file)
## group
dge.list <- re.sample.group(dge.list, meta.df)
## annotation
dge.list <- anno.into.list(dge.list, gene.anno, "ensembl_gene_id")
## -------------------------------------
group. <- dge.list$samples$group
## design
design <- model.matrix(~ 0 + group.)
## -------------------------------------
## contrast
contr.matrix <- limma::makeContrasts(
  treat_4.vs.contr = group.ahr_lna_4_6 - group.nc_lna_6,
  treat_7.vs.contr = group.ahr_lna_7_6 - group.nc_lna_6,
  levels = design
)
```

```
## ---------------------------------------
res <- limma_downstream(dge.list, group., design, contr.matrix)
## ---------------------------------------
## ========== Run block ==========
mapply(res, paste0(names(res), "_results.tsv"), FUN = write_tsv)
```

27 File: from_dataset.R

```
## R
## metadata
## path <- "~/operation/geo_db/ahr_sig"
## ----------------------------------------------------------------------
## ========== Run block ==========
meta <- data.table::fread("series.csv") %>%
  dplyr::filter(grepl("Expression", `Series Type`),
                grepl("Homo sapiens", Taxonomy)) %>%
  dplyr::as_tibble()
## ----------------------------------------------------------------------
## Use wget to download data
apply(dplyr::mutate(meta, seq = 1:nrow(meta)), 1,
      function(vec){
        cat("[Info] Downloading seq of", vec[["seq"]], "\n")
        gse <- vec[["Accession"]]
        gse.dir <- gsub("[0-9]{3}$", "nnn", gse)
        ftp <- paste0("ftp://ftp.ncbi.nlm.nih.gov/geo/series/", gse.dir, "/", gse, "/suppl/")
        system(paste("wget -np -m", ftp))
      })
## ----------------------------------------------------------------------
## test for download data
## ---------------------------------------
# file <- list.files(pattern = test)
# info.t <- GEOquery::getGEO(filename = file)
# ## ----------------------------------------------------------------------
# df <- Biobase::phenoData(info.t)
# name.sample <- Biobase::sampleNames(df)
# ## ---------------------------------------
# sample <- GEOquery::getGEO(name.sample[1])
# sample.df <- GEOquery::Table(sample)
# ## ----------------------------------------------------------------------
```

28 File: gather_results.R

```r
## R
## gather analysis results
setwd("~/operation/geo_db/ahr_sig/")
## -------------------------------------
all_results <- list.files(pattern = "_results.tsv",
                          recursive = T,
                          full.names = T) %>%
  data.frame() %>%
  dplyr::rename(file = 1) %>%
  dplyr::mutate(filename = stringr::str_extract(file, "[^/]*$"),
                contrast = stringr::str_extract(filename, "^.*(?=_results)"),
                series = stringr::str_extract(file, "(?<=/)GSE[0-9]{1,}(?=/)"))
## ----------------------------------------------------------------------
## cell, treat.left, treat.right
all_series <- all_results$series %>%
  unique()
## ----------------------------------------------------------------------
lst.sum <- lapply(all_series, function(gse){
                    info <- try_do("GEOquery::getGEO(gse)", envir = environment())
                    info <- lapply(info,
                                   function(obj){
                                     obj <- Biobase::experimentData(obj)
                                     obj@other$summary
                                   })
                })
names(lst.sum) <- all_series
## ----------------------------------------------------------------------
lst.sum <- lapply(lst.sum, `[`, 1) %>%
  data.table::data.table() %>%
  dplyr::rename(summary = 1) %>%
  dplyr::mutate(Accession = all_series, summary = unlist(summary)) %>%
  dplyr::as_tibble()
## ----------------------------------------------------------------------
meta.summary <- merge(meta, lst.sum, by = "Accession", all.y = T) %>%
  dplyr::as_tibble()
write_tsv(meta.summary, "meta.summary.tsv")
## ----------------------------------------------------------------------
gene.anno <- anno.gene.biomart("hsapiens_gene_ensembl",
                               attr = c("ensembl_gene_id", "hgnc_symbol"))
## ----------------------------------------------------------------------
```

```r
screened.genes <- lapply(all_results$file, data.table::fread) %>%
  lapply(function(df){
           df <- df %>%
             dplyr::rename(ensembl = 1, symbol = 2)
           ## if number > n, filter out the rest
           n <- 1000
           if(nrow(df) > n){
             ## col of adjust p.value
             adj.p <- colnames(df) %>%
               .[grepl("adj|q-value", ., ignore.case = T)]
             df <- dplyr::rename(df, adj.p = paste0(adj.p)) %>%
               dplyr::arrange(adj.p) %>%
               ## get top n
               head(n = n) %>%
               dplyr::relocate(ensembl, symbol)
           }
           df <- dplyr::select(df, 1:2)
           return(df)
                }) %>%
  data.table::rbindlist() %>%
  dplyr::filter(!is.na(ensembl) & !is.na(symbol) & symbol != "") %>%
  dplyr::filter(ensembl %in% all_of(gene.anno$ensembl_gene_id)) %>%
  # dplyr::mutate(symbol = gsub("\\.[0-9]$", "", symbol)) %>%
  dplyr::distinct() %>%
  dplyr::as_tibble()
## ----------------------------------------------------------------------
## AHR targets were retrieved from the Transcription Factor Target Gene Database
## <http://tfbsdb.systemsbiology.net/>
tf.db <- data.table::fread("TFTGD_ahr_targes.tsv") %>%
  dplyr::mutate(symbol = ifelse(grepl("^V_", Motif),
                                stringr::str_extract(Motif, "(?<=^V_)[^_]{1,}"),
                                stringr::str_extract(Motif, "^[^_]{1,}"))) %>%
  dplyr::distinct(symbol)
## -------------------------------------
tf.db <- dplyr::filter(gene.anno, hgnc_symbol %in% all_of(tf.db$symbol)) %>%
  dplyr::select(ensembl_gene_id, hgnc_symbol) %>%
  distinct() %>%
  dplyr::rename(ensembl = 1, symbol = 2)
## ----------------------------------------------------------------------
## ========== Run block ==========
## merge genes from 'screened.genes' and 'tf.db'
```

```r
merge.scre_tf <- dplyr::bind_rows(screened.genes, tf.db) %>%
  dplyr::distinct()
write_tsv(merge.scre_tf, "merge.scre_tf.tsv")
## -----------------------------------------------------------------------
```