Reproducible multiple sample script

Selcen Ari Alper Yilmaz

2021-02-19

Generating networks for multiple sample

This file contains an example of competing endogenous RNA analysis with ceRNAnetsim package.

1. Downloading data for analysis

```
#query for Ovarian and Breast Carcinoma in TCGA. In this example TCGAbiolinks package was used.

query_gene_exp <- GDCquery(project = c("TCGA-BRCA", "TCGA-OV"),
                  data.category = "Transcriptome Profiling",
                  data.type = "Gene Expression Quantification",
                  workflow.type = "HTSeq - Counts")


query_mirna_exp <- GDCquery(project = c("TCGA-BRCA", "TCGA-OV"),
                  data.category = "Transcriptome Profiling",
                  data.type = "Isoform Expression Quantification",
                  workflow.type = "BCGSC miRNA Profiling")

#finding samples which have both miRNA and gene expression dataset


getResults(query_gene_exp, cols = c("project","cases"))%>%
  mutate(barcode = substr(cases, 1,12))-> gene_exp_cases

getResults(query_mirna_exp, cols = c("project","cases"))%>%
  mutate(barcode = substr(cases, 1,12))-> mirna_exp_cases

# For demostration purposes we select 10 random patients per cancer project.
set.seed(1234)
gene_exp_cases%>%
  inner_join(mirna_exp_cases, by = c("project", "barcode"))%>%
  dplyr::select(project, barcode)%>%
  distinct()%>%
  group_by(project)%>%
  sample_n(10)%>%
  pull(barcode)->selected_barcodes
```

query for selected patient barcodes

```
query_gene_exp_selected <- GDCquery(project = c("TCGA-BRCA", "TCGA-OV"),
                  data.category = "Transcriptome Profiling",
                  data.type = "Gene Expression Quantification",
                  workflow.type = "HTSeq - Counts",
                  barcode = selected_barcodes)


query_mirna_exp_selected <- GDCquery(project = c("TCGA-BRCA", "TCGA-OV"),
                  data.category = "Transcriptome Profiling",
                  data.type = "Isoform Expression Quantification",
                  workflow.type = "BCGSC miRNA Profiling",
                  barcode = selected_barcodes)
```

Downloading and preparing data

```
GDCdownload(query_gene_exp_selected)
GDCdownload(query_mirna_exp_selected)

gene_exps <- GDCprepare(query_gene_exp_selected)
```

```
## |                                                  |   0%                      |==
```

```
mirna_exps <- GDCprepare(query_mirna_exp_selected)
```

```
##   |                                                                            |
```

Preparing gene expression dataset

```
#preparation of gene expression dataset

as.data.frame(assay(gene_exps))%>%
  mutate(ensembl_gene_id = rownames(.))%>%
  dplyr::left_join(as.data.frame(rowData(gene_exps)),
                   by = "ensembl_gene_id")%>%
  dplyr::select(Ensembl_Gene_Id = ensembl_gene_id, external_gene_name, 1:21)->gene_exp_to_be_analyzed

#fixing sample ids
cases <- which(substring(names(gene_exp_to_be_analyzed),1,4) %in% "TCGA")
names(gene_exp_to_be_analyzed)[cases] <- substr(names(gene_exp_to_be_analyzed)[cases], 1,12)

head(gene_exp_to_be_analyzed)
```

```
##   Ensembl_Gene_Id external_gene_name TCGA-GM-A3NW TCGA-A2-A0YT TCGA-S3-AA17
## 1 ENSG00000000003             TSPAN6         4211         1576         3454
## 2 ENSG00000000005               TNMD           20            2            1
```

```
## 3 ENSG00000000419              DPM1         1772         4163         2355
## 4 ENSG00000000457             SCYL3         2142         3669         1378
## 5 ENSG00000000460          C1orf112          922          834          642
## 6 ENSG00000000938               FGR          473          369         1124
##   TCGA-E2-A1B6 TCGA-BH-A0AY TCGA-BH-A0AY TCGA-A7-A5ZW TCGA-A2-A0EM TCGA-E2-A154
## 1         2943         2418         4451         5904         6760         4541
## 2           23           62          660            9           13            0
## 3         4172         1675         1717         2067         1508         1391
## 4         1983         1460         1343         2204         2460         2004
## 5          745          406          314          510         1124          688
## 6         2274          350          366          370          417          141
##   TCGA-D8-A1Y2 TCGA-GM-A2DL TCGA-09-2045 TCGA-04-1341 TCGA-24-1842 TCGA-24-2020
## 1          714         1268         2174         4046         1784        12391
## 2            8          121            0            8            0            3
## 3         2375         1538         1158         3543         3781         6350
## 4         1115         1061          835          630          923         1423
## 5          481          360          150          396         1027         2395
## 6         2594          612          155          210         1085          266
##   TCGA-13-0725 TCGA-25-2392 TCGA-04-1347 TCGA-04-1365 TCGA-13-1408 TCGA-13-0924
## 1         5121         7594         5659         6037         3731         3149
## 2           10            6           13            6            2            0
## 3         4044         3590         5126         4280         4138         2190
## 4          303         1410          537          944         1220          287
## 5          420          634          272          896          760          392
## 6          291         1055          103          866          330          328
```

Preparing miRNA expression dataset

```
# miRNA expression dataset contains miRNAs with mirbase ids. So, firstly, mirbase_id_coversion dataset

mirbase_url <- "ftp://mirbase.org/pub/mirbase/21/genomes/hsa.gff3"

read_tsv(mirbase_url, comment = "#", col_names = FALSE) %>%
  dplyr::select(mirna_type= X3, definition = X9)%>%
  filter(!endsWith(mirna_type, "primary_transcript"))%>%
  tidyr::separate(definition, c("ID", "Alias", "Name", "Derivated"), sep = ";")%>%
  dplyr::select(Alias, Name)%>%
  tidyr::separate(Alias, c("trash1", "ID"), sep = "=")%>%
  tidyr::separate(Name, c("trash2", "Name"), sep = "=")%>%
  dplyr::select(-trash1, -trash2)-> mirbase_id_conv

# preparation of miRNA expression dataset.
# We used miRBase (Version 21) to obtain miRBase id (like MIMAT0000) for each mature isoform and
# aggregated readr per million for each isoform.

mirna_exps%>%
  as.data.frame()%>%
  dplyr::select(miRNA_ID,
                read_count,
                reads_per_million_miRNA_mapped,
                miRNA_region,
                barcode)%>%
```

```
  dplyr::filter(startsWith(miRNA_region, "mature"))%>%
  dplyr::mutate(mirbase_id =str_remove(miRNA_region, "mature,"))%>%
  dplyr::select(-miRNA_region)%>%
  dplyr::inner_join(mirbase_id_conv,
                    by = c("mirbase_id"="ID"))%>%
  dplyr::group_by(Name, barcode)%>%
  mutate(read_count= sum(read_count),
         reads_per_million_miRNA_mapped = sum(reads_per_million_miRNA_mapped))%>%
  dplyr::ungroup()%>%
  dplyr::select(miRNA = Name, reads_per_million_miRNA_mapped, barcode)%>% # reads_per_million_miRNA_mapp
  distinct()%>%
  tidyr::pivot_wider(names_from = "barcode", values_from = "reads_per_million_miRNA_mapped")-> mirna_exp

#fixing sample ids
cases <- which(substring(names(mirna_exp_to_be_analyzed),1,4) %in% "TCGA")
names(mirna_exp_to_be_analyzed)[cases] <- substr(names(mirna_exp_to_be_analyzed)[cases], 1,12)

head(mirna_exp_to_be_analyzed)
```

```
## # A tibble: 6 x 22
##   miRNA `TCGA-GM-A3NW` `TCGA-A2-A0YT` `TCGA-S3-AA17` `TCGA-E2-A1B6`
##   <chr>          <dbl>          <dbl>          <dbl>          <dbl>
## 1 hsa-~     176159.          58217.         125433.         120453.
## 2 hsa-~        117.             91.6           38.0            96.4
## 3 hsa-~          0.443           8.52           1.79           15.7
## 4 hsa-~      38402.          21955.         10537.          24443.
## 5 hsa-~         40.1            35.2           30.8            83.1
## 6 hsa-~       2555.           1112.           435.           4391.
## # ... with 17 more variables: `TCGA-BH-A0AY` <dbl>, `TCGA-BH-A0AY` <dbl>,
## #   `TCGA-A7-A5ZW` <dbl>, `TCGA-A2-A0EM` <dbl>, `TCGA-E2-A154` <dbl>,
## #   `TCGA-D8-A1Y2` <dbl>, `TCGA-GM-A2DL` <dbl>, `TCGA-09-2045` <dbl>,
## #   `TCGA-04-1341` <dbl>, `TCGA-24-1842` <dbl>, `TCGA-24-2020` <dbl>,
## #   `TCGA-13-0725` <dbl>, `TCGA-25-2392` <dbl>, `TCGA-04-1347` <dbl>,
## #   `TCGA-04-1365` <dbl>, `TCGA-13-1408` <dbl>, `TCGA-13-0924` <dbl>
```

Performing competing endogenous RNA (ceRNAnetsim) analysis

Firstly, miRNA:gene pair dataset must be defined. Here, dataset of miRNA:gene pairs which were obtained from high-throughput experimental studies is used as an example. Note that in manuscript SPONGE analysis was preformed to refine miRNA:gene pairs. In this demonstration we omitted SPONGE analysis and use bare

```
experimentalmirnagene <- readRDS("data/experimentalmirnagene.RDS")

graph_list = list()

for(i in selected_barcodes){

current_network <- experimentalmirnagene%>%
  right_join(dplyr::select(gene_exp_to_be_analyzed, Ensembl_Gene_Id, i), by="Ensembl_Gene_Id")%>%
  right_join(dplyr::select(mirna_exp_to_be_analyzed, miRNA, i), by ="miRNA", suffix= c("Gene_expression
```

```
  dplyr::select(Ensembl_Gene_Id, miRNA, Gene_expression= paste0(i, "Gene_expression"), miRNA_expression
  filter(!is.na(Gene_expression), !is.na(miRNA_expression))%>%
  filter(Gene_expression != 0, miRNA_expression != 0)%>%
  priming_graph(competing_count = Gene_expression,
                miRNA_count = miRNA_expression)

graph_list[[i]] <- current_network

}
```

Now we have miRNA:gene networks for 20 patients in a list. So any of them can be used with various functions provided by ceRNAnetsim package. Below we just printing out network of a single patient.

```
graph_list$`TCGA-S3-AA17`
```

```
## # A tbl_graph: 8630 nodes and 27440 edges
## #
## # A directed acyclic simple graph with 10 components
## #
## # Node Data: 8,630 x 7 (active)
##   name     type   node_id initial_count count_pre count_current changes_variable
##   <chr>    <chr>    <int>         <dbl>     <dbl>         <dbl> <chr>
## 1 ENSG000~ Compe~       1          2959      2959          2959 Competing
## 2 ENSG000~ Compe~       2          7049      7049          7049 Competing
## 3 ENSG000~ Compe~       3          1717      1717          1717 Competing
## 4 ENSG000~ Compe~       4          4738      4738          4738 Competing
## 5 ENSG000~ Compe~       5          1028      1028          1028 Competing
## 6 ENSG000~ Compe~       6          1925      1925          1925 Competing
## # ... with 8,624 more rows
## #
## # Edge Data: 27,440 x 19
##    from    to Competing_name miRNA_name Gene_expression miRNA_expression dummy
##   <int> <int> <chr>          <chr>                <dbl>            <dbl> <dbl>
## 1     1  8266 ENSG000001133~ hsa-let-7~            2959          125433.     1
## 2     2  8266 ENSG000001006~ hsa-let-7~            7049          125433.     1
## 3     3  8266 ENSG000000805~ hsa-let-7~            1717          125433.     1
## # ... with 27,437 more rows, and 12 more variables: afff_factor <dbl>,
## #   degg_factor <dbl>, comp_count_list <list>, comp_count_pre <dbl>,
## #   comp_count_current <dbl>, mirna_count_list <list>, mirna_count_pre <dbl>,
## #   mirna_count_current <dbl>, mirna_count_per_dep <dbl>, effect_current <dbl>,
## #   effect_pre <dbl>, effect_list <list>
```