Building protein networks around drug-targets using OmnipathR

true true true

Abstract

Many applications require to connect drugs to proteins in signaling networks. OmnipathR provides easy access to curated pathway resources from OmniPath. Here we use data from DrugBank to find direct protein targets of drugs and to connect them to downstream signaling proteins using OmnipathR.

Introduction

In many applications we would like to understand how a specific drug interacts with the protein signaling network through its targets.

```
library(dplyr)
library(ggplot2)
library(OmnipathR)
library(igraph)
library(ggraph)
```

Initialise OmniPath database

We query protein-protein interactions from the webservice of OmniPath [1,2] at https://omnipathdb.org/ using OmnipathR package:

```
# Download protein-protein interactions
interactions = import_omnipath_interactions() %>% as_tibble()
```

```
## Downloaded 41419 interactions.
```

```
# Convert to igraph objects:
OPI_g = interaction_graph(interactions = interactions )
```

Querying drug targets

For direct drug targets we will use DrugBank [3] database accessed via the *dbparser* package. Please note, that the following few chuncks of code is not evaluated. DrugBank requires registrations to access the data, therefore we ask the reader to register at DrugBank and download the data from here.

The next block of code is used to process the DrugBank dataset.

```
library(dbparser)
library(XML)


## parse data from XML and save it to memory
get_xml_db_rows("..path-to-DrugBank/full database.xml")

## load drugs data
```

```
drugs <- parse_drug() %>% select(primary_key, name)
drugs <- rename(drugs,drug_name = name)

## load drug target data
drug_targets <- parse_drug_targets() %>%
   select(id, name,organism,parent_key) %>%
   rename(target_name = name)

## load polypeptide data
drug_peptides <- parse_drug_targets_polypeptides()  %>%
   select(id, name, general_function, specific_function,
          gene_name, parent_id) %>%
   rename(target_name = name, gene_id = id)

# join the 3 datasets
drug_targets_full <- inner_join(drug_targets, drug_peptides,
                                by=c("id"="parent_id", "target_name")) %>%
   inner_join(drugs, by=c("parent_key"="primary_key")) %>%
   select(-other_keys)
```

Here we declare the names of drugs of interest.

```
drug_names = c("Valproat"      = "Valproic Acid",
               "Diclofenac"    = "Diclofenac",
               "Paracetamol"   = "Acetaminophen",
               "Ciproflaxin"   = "Ciprofloxacin",
               "Nitrofurantoin"= "Nitrofurantoin",
               "Tolcapone",
               "Azathioprine",
               "Troglitazone",
               "Nefazodone",
               "Ketoconazole",
               "Omeprazole",
               "Phenytoin",
               "Amiodarone",
               "Cisplatin",
               "Cyclosporin A"  = "Cyclosporine",
               "Verapamil",
               "Buspirone",
               "Melatonin",
               "N-Acetylcysteine"= "Acetylcysteine",
               "Vitamin C"       = "Ascorbic acid",
               "Famotidine",
               "Vancomycin")
```

```
drug_target_data_sample <- drug_targets_full %>%
   filter(organism == "Humans",drug_name %in% drug_names)
```

We only use a small sample of the database:

```
drug_targets <- OmnipathR:::drug_target_data_sample %>%
   filter(organism == "Humans",drug_name %in% drug_names)
```

Quality control

Check which drug targets are in Omnipath

```
drug_targets <-  drug_targets %>%
   select(-target_name, -organism) %>%
   mutate(in_OP = gene_id %in% c(interactions$source))
# not all drug-targets are in OP.
print(all(drug_targets$in_OP))
```

```
## [1] FALSE
```

```
# But each drug has at least one target in OP.
drug_targets %>% group_by(drug_name) %>% summarise(any(in_OP))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 19 x 2
##    drug_name      `any(in_OP)`
##    <chr>          <lgl>
##  1 Acetaminophen  TRUE
##  2 Acetylcysteine TRUE
##  3 Amiodarone     TRUE
##  4 Ascorbic acid  TRUE
##  5 Azathioprine   TRUE
##  6 Buspirone      TRUE
##  7 Ciprofloxacin  FALSE
##  8 Cisplatin      TRUE
##  9 Diclofenac     TRUE
## 10 Famotidine     TRUE
## 11 Ketoconazole   TRUE
## 12 Melatonin      TRUE
## 13 Nefazodone     TRUE
## 14 Omeprazole     TRUE
## 15 Phenytoin      TRUE
## 16 Tolcapone      TRUE
## 17 Troglitazone   TRUE
## 18 Valproic Acid  TRUE
## 19 Verapamil      TRUE
```

Downstream signaling nodes

We would like to investigate the effect of the drugs on some selected proteins. For example, the activity of these proteins are measured upon the drug perturbation. We'll build a network from the drug targets to these selected nodes.

First we declare protein of interest (POI):

```
POI = tibble(protein = c("NFE2L2","HMOX1","TP53","CDKN1A","BTG2","NFKB1",
                         "ICAM1","HSPA5", "ATF4","DDIT3","XBP1"))
```

Quality control

Checking which POI are in Omnipath

```
POI <- POI %>% mutate(in_OP = protein %in% interactions$target_genesymbol)
# all POI is in Omnipath
```

```
print(all(POI$in_OP))
```

```
## [1] TRUE
```

Build network between drug targets and POI

First, we find paths between the drug targets and the POIs. For the sake of this simplicity we focus on drug targets of one drug, *Cisplatin*.

The paths are represented by a set of nodes:

```
source_nodes <- drug_targets %>%
   filter(in_OP, drug_name=="Cisplatin") %>%
   pull(gene_name)
target_nodes <- POI %>% filter(in_OP) %>% pull(protein)

collected_path_nodes = list()

for(i_source in 1:length(source_nodes)){

   paths <- shortest_paths(OPI_g, from = source_nodes[[i_source]],
                           to = target_nodes,
                           output = 'vpath')
   path_nodes <- lapply(paths$vpath,names) %>% unlist() %>% unique()
   collected_path_nodes[[i_source]] <- path_nodes
}
collected_path_nodes <- unlist(collected_path_nodes) %>% unique()
```

The direct drug targets, the POIs and the intermediate pathway members give rise to the network.

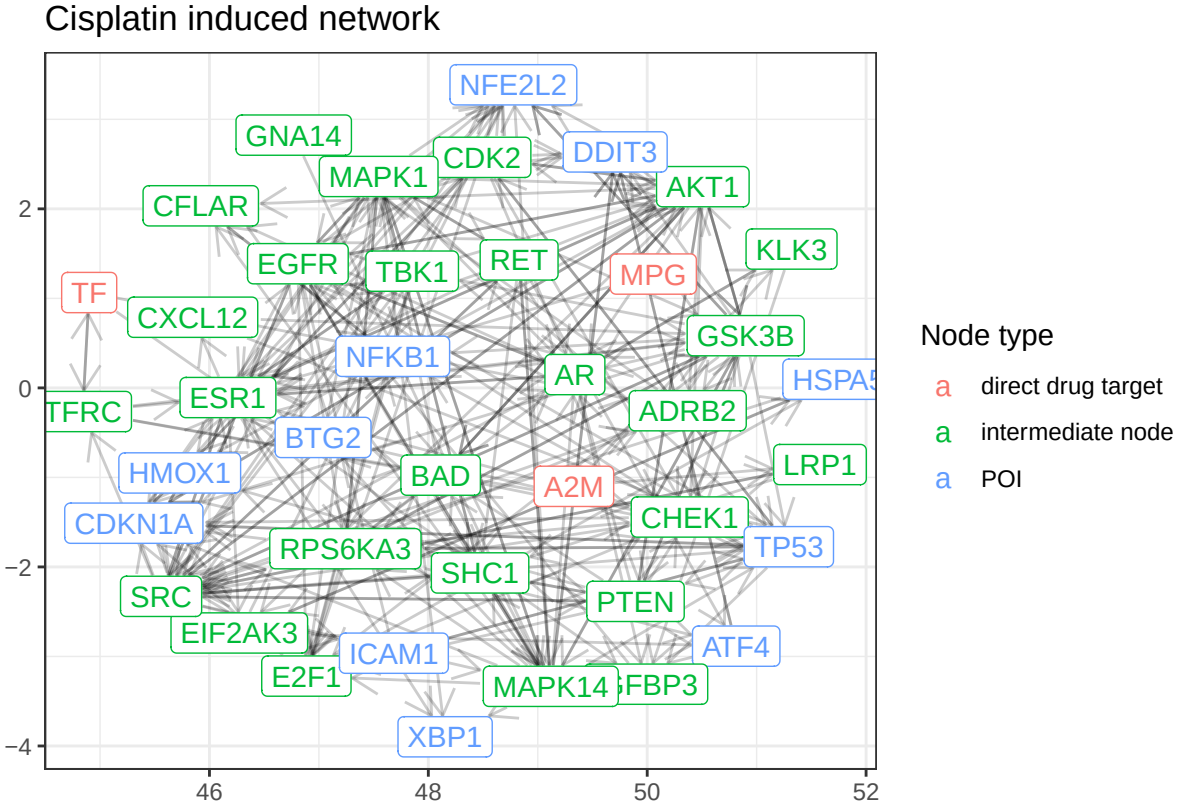
```
cisplatin_nodes <- c(source_nodes,target_nodes, collected_path_nodes) %>%
   unique()
cisplatin_network <- induced_subgraph(graph = OPI_g,vids = cisplatin_nodes)
```

We annotate the nodes of the network and plot it.

```
V(cisplatin_network)$node_type = ifelse(
   V(cisplatin_network)$name %in% source_nodes, "direct drug target",
   ifelse(
      V(cisplatin_network)$name %in% target_nodes,"POI","intermediate node"))

ggraph(
      cisplatin_network,
      layout = "lgl",
      area = vcount(cisplatin_network)^2.3,
      repulserad = vcount(cisplatin_network)^1.2,
      coolexp = 1.1
   ) +
   geom_edge_link(
      aes(
         start_cap = label_rect(node1.name),
         end_cap = label_rect(node2.name)),
         arrow = arrow(length = unit(4, 'mm')
      ),
      edge_width = .5,
```

```
    edge_alpha = .2
) +
geom_node_point() +
geom_node_label(aes(label = name, color = node_type)) +
scale_color_discrete(
    guide = guide_legend(title = 'Node type')
) +
theme_bw() +
xlab("") +
ylab("") +
ggtitle("Cisplatin induced network")
```



The above network represents a way how Cisplatin can influence the POIs. One can for example filter out edges based on the number fo resources reporting the edge or based on the number of papers mentioning it. However, this is already covered by previous pypath tutorials.

Acknowledgements

The above pipeline was inspired by the post of Denes Turei available here.

References

[1] D Turei, A Valdeolivas, L Gul, N Palacio-Escat, O Ivanova, A Gabor, D Modos, T Korcsmaros and J Saez-Rodriguez (2020) Integrated intra- and intercellular signaling knowledge for multicellular omics analysis. *bioRxiv* 2020.08.03.221242

[2] D Turei, T Korcsmaros and J Saez-Rodriguez (2016) OmniPath: guidelines and gateway for literature-curated signaling pathway resources. *Nature Methods* 13(12)

[3] Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, Sajed T, Johnson D, Li C, Sayeeda Z, Assempour N, Iynkkaran I, Liu Y, Maciejewski A, Gale N, Wilson A, Chin L, Cummings R, Le D, Pon A, Knox C, Wilson M. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* 2017 Nov 8. doi: 10.1093/nar/gkx1037.

Session info

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Arch Linux
##
## Matrix products: default
## BLAS:   /usr/lib/libblas.so.3.9.0
## LAPACK: /usr/lib/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] rmarkdown_2.4    BiocStyle_2.16.1 ggraph_2.0.3     OmnipathR_1.99.5
##  [5] jsonlite_1.7.1   igraph_1.2.5     ggplot2_3.3.2    dplyr_1.0.2
##  [9] knitr_1.30       colorout_1.2-2
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.5          highr_0.8           BiocManager_1.30.10
##  [4] pillar_1.4.6        compiler_4.0.3      viridis_0.5.1
##  [7] tools_4.0.3         digest_0.6.25       viridisLite_0.3.0
## [10] evaluate_0.14       lifecycle_0.2.0     tibble_3.0.3
## [13] gtable_0.3.0        pkgconfig_2.0.3     rlang_0.4.7
## [16] tidygraph_1.2.0     cli_2.0.2           yaml_2.2.1
## [19] ggrepel_0.8.2       xfun_0.18           gridExtra_2.3
## [22] withr_2.3.0         stringr_1.4.0       graphlayouts_0.7.0
## [25] generics_0.0.2      vctrs_0.3.4         grid_4.0.3
## [28] tidyselect_1.1.0    glue_1.4.2          R6_2.4.1
## [31] fansi_0.4.1         bookdown_0.20       polyclip_1.10-0
## [34] purrr_0.3.4         tidyr_1.1.2         tweenr_1.0.1
## [37] farver_2.0.3        magrittr_1.5        htmltools_0.5.0
## [40] scales_1.1.1        ellipsis_0.3.1      MASS_7.3-53
## [43] assertthat_0.2.1    ggforce_0.3.2       colorspace_1.4-1
## [46] labeling_0.3        tinytex_0.26        utf8_1.1.4
## [49] stringi_1.5.3       munsell_0.5.0       crayon_1.3.4
```