NicheNet: Assess how well top-ranked ligands can predict a gene set of interest ==========
Alberto Valdeolivas: alberto.valdeolivas@bioquant.uni-heidelberg.de; Date: 23/06/2020

Abstract

This vignette aims at reproducing the results of the following original vignette, available in the NicheNetr repository:

https://github.com/saeyslab/nichenetr/blob/master/vignettes/target_prediction_evaluation_geneset.md

In our particular case, we use sets of interactions availabe in the **Omnipath** database. We will study potential ligand-targets influence upon SARS-CoV-2 infection.

Introduction

This vignette shows how NicheNet can be used to to predict which ligands might regulate a given set of genes and how well they do this prediction. For this analysis, one needs to define:

- a set of genes of which expression in a "receiver cell" is possibly affected by extracellular protein signals (ligands) (e.g. genes differentially expressed upon cell-cell interaction)
- a set of potentially active ligands (e.g. ligands expressed by interacting "sender cells")

Therefore, you often first need to process expression data of interacting cells to define both.

In this example, we are going to use expression data after SARS-CoV-2 infection to try to dissect which ligands expressed by infected cells can have an i nfluence on the expression of target genes in the same cell lines (Autocrine view). In particular, we will focus on the inflammation response potentially induced by this ligands.

```
library(nichenetr)
library(tidyverse)
library(fgsea)
```

Read expression data of interacting cells

First, we read the results of the differentially expression analysis after infection with SARS-CoV-2 on the CALU-3 cell line.

```
expressed_genes_receiver <-
    readRDS("Results/dds_results_CALU3vsCOV2.rds") %>%
    as.data.frame() %>%
    tibble::rownames_to_column(var = "Gene") %>%
    dplyr::filter(!is.na(stat)) %>%
    dplyr::pull(Gene)
```

Secondly, we determine which ligands are over-expressed after SARS-CoV-2 infection.

```
padj_tres <- 0.1
log2FoldChange_tres <- 1

## We take our ligands in the network
ligands <-
    readRDS("OmniNetworks_NNformat/lr_Network_Omnipath.rds") %>%
    dplyr::pull(from) %>%
    unique()

DDS_CALU3_ligands <-
    readRDS("Results/dds_results_CALU3vsCOV2.rds") %>%
    as.data.frame() %>%
```

```
    tibble::rownames_to_column(var = "Gene") %>%
    dplyr::filter(padj < padj_tres,
                  log2FoldChange > log2FoldChange_tres,
                  Gene %in% ligands) %>%
    dplyr::pull(Gene)
```

Load the ligand-target model we want to use

```
ligand_target_matrix <- readRDS("Results/ligand_target_matrixWithweights.rds")
ligand_target_matrix[1:5,1:5] # target genes in rows, ligands in columns
##               CALM1        WNT5A       CXCL16       CCL3L3      TNFSF10
## A1BG   0.0000000000 0.0000000000 0.000000e+00 0.000000e+00 0.0000000000
## A1CF   0.0000000000 0.0000000000 0.000000e+00 0.000000e+00 0.0000000000
## A2M    0.0011027517 0.0004845514 2.936421e-03 5.441192e-03 0.0017391820
## A2ML1  0.0000000000 0.0000000000 0.000000e+00 0.000000e+00 0.0000000000
## A4GALT 0.0002105736 0.0001070804 5.825834e-05 9.488076e-05 0.0001410451
```

Load the gene set of interest and background of genes

To establish a gene set of interest, we perform a Gene set Enrichment analysis (GSEA) and we check among the most appealing overrepresanted signatures upon SARS-CoV-2 infection. We remove the differentially expressed ligands from this comparison.

```
ranks <- readRDS("Results/dds_results_CALU3vsCOV2.rds") %>%
    as.data.frame() %>%
    tibble::rownames_to_column(var = "Gene") %>%
    dplyr::filter(!(Gene %in% DDS_CALU3_ligands)) %>%
    dplyr::filter(!is.na(stat)) %>%
    dplyr::pull(stat, name=Gene)

# immunologic_signatures <- gmtPathways("NicheNet_Omnipath/c7.all.v7.1.symbols.gmt")
hallmarlk_signatures <- gmtPathways("h.all.v7.1.symbols.gmt")
# go_signatures <- gmtPathways("NicheNet_Omnipath/c5.bp.v7.1.symbols.gmt")

fgseaRes <- fgsea(hallmarlk_signatures, ranks,  nperm=1000)
# Testing only one pathway is implemented in a more efficient manner

SignificantResults <- fgseaRes %>%
    dplyr::filter(padj < 0.01) %>%
    dplyr::arrange(desc(NES)) %>%
    dplyr::top_n(12, abs(NES))
SignificantResults
##                              pathway        pval        padj         ES
##  1: HALLMARK_INTERFERON_GAMMA_RESPONSE 0.001369863 0.005056634  0.8627654
##  2:   HALLMARK_TNFA_SIGNALING_VIA_NFKB 0.001386963 0.005056634  0.8608318
##  3: HALLMARK_INTERFERON_ALPHA_RESPONSE 0.001550388 0.005056634  0.9172465
##  4:     HALLMARK_INFLAMMATORY_RESPONSE 0.001440922 0.005056634  0.7274370
##  5:   HALLMARK_IL6_JAK_STAT3_SIGNALING 0.001579779 0.005056634  0.7126008
##  6:                   HALLMARK_HYPOXIA 0.001375516 0.005056634  0.5893036
##  7:                HALLMARK_APOPTOSIS 0.001438849 0.005056634  0.5743182
##  8:            HALLMARK_G2M_CHECKPOINT 0.003831418 0.006561680 -0.5370578
##  9:            HALLMARK_MYC_TARGETS_V2 0.002638522 0.006561680 -0.6827875
## 10:            HALLMARK_MYC_TARGETS_V1 0.003921569 0.006561680 -0.6785459
## 11:               HALLMARK_E2F_TARGETS 0.003937008 0.006561680 -0.6829123
```

```
## 12: HALLMARK_OXIDATIVE_PHOSPHORYLATION 0.003787879 0.006561680 -0.6946604
##           NES nMoreExtreme size                                leadingEdge
##  1:  3.143190            0  174     OAS2,IFIT1,RSAD2,IFIT2,IFIT3,TNFAIP3,...
##  2:  3.114605            0  161 IFIT2,TNFAIP3,ATF3,PPP1R15A,NFKBIA,IFIH1,...
##  3:  3.029033            0   89        RSAD2,IFIT2,IFIT3,MX1,IFIH1,TXNIP,...
##  4:  2.590435            0  148       NFKBIA,IRF1,LAMP3,IFITM1,KLF6,RTP4,...
##  5:  2.269832            0   67         IRF1,STAT2,MAP3K8,STAT1,JUN,PIM1,...
##  6:  2.144207            0  173  TNFAIP3,ATF3,PPP1R15A,TIPARP,DUSP1,STC2,...
##  7:  2.039701            0  140       ATF3,TXNIP,IRF1,TAP1,PMAIP1,ISG20,...
##  8: -2.237533            0  190        KPNA2,MCM5,SQLE,HSPA8,MCM6,LMNB1,...
##  9: -2.354863            0   58        TMEM97,MCM5,PHB,DCTPP1,PLK1,MCM4,...
## 10: -2.827153            0  193          KPNA2,MCM5,PHB,MCM6,SRSF2,NME1,...
## 11: -2.857211            0  195       KPNA2,MCM5,MXD3,SPAG5,NCAPD2,POLD1,...
## 12: -2.878447            0  184      MAOB,POLR2F,COX8A,LDHB,VDAC3,NDUFB2,...
```

I select inflammation related genes.

```
## I am going to check with inflammation genes
inflammationGenes <- SignificantResults %>%
    dplyr::filter(pathway == "HALLMARK_INFLAMMATORY_RESPONSE") %>%
    dplyr::pull(leadingEdge) %>% unlist()

## We chech that there are no upregulated ligands here.
intersect(DDS_CALU3_ligands,inflammationGenes )
## character(0)

geneset_oi <- inflammationGenes[inflammationGenes %in% rownames(ligand_target_matrix)]

head(geneset_oi)
## [1] "NFKBIA" "IRF1"   "IFITM1" "KLF6"   "RTP4"   "IRAK2"
background_expressed_genes <- expressed_genes_receiver %>%
    .[. %in% rownames(ligand_target_matrix)]
head(background_expressed_genes)
## [1] "SAMD11"   "NOC2L"    "ISG15"    "AGRN"     "TNFRSF18" "SDF4"
```

Perform NicheNet's ligand activity analysis on the gene set of interest

As potentially active ligands, we will use ligands that are 1) Over-expressed in CALU3 after SARS-CoV-2 infection and 2) can bind a (putative) receptor expressed by malignant cells. Putative ligand-receptor links were gathered from Omnipath ligand-receptor data sources.

```
expressed_ligands <- intersect(ligands,DDS_CALU3_ligands)

receptors <- unique(lr_network$to)
expressed_receptors <- intersect(receptors,expressed_genes_receiver)

potential_ligands <- lr_network %>%
    filter(from %in% expressed_ligands & to %in% expressed_receptors) %>%
    pull(from) %>%
    unique()
head(potential_ligands)
## [1] "CXCL1" "CXCL2" "CXCL3" "CXCL5" "CCL20" "CCL17"
```

we now calculate the ligand activity of each ligand, or in other words, we will assess how well each over-expressed ligand after viral infection can predict the inflammation gene set compared to the background of

expressed genes (predict whether a gene belongs to the inflammation program or not).

```
ligand_activities <- predict_ligand_activities(
    geneset = geneset_oi,
    background_expressed_genes = background_expressed_genes,
    ligand_target_matrix = ligand_target_matrix,
    potential_ligands = potential_ligands)
```

Now, we want to rank the ligands based on their ligand activity. In our validation study, we showed that the pearson correlation between a ligand's target predictions and the observed transcriptional response was the most informative measure to define ligand activity. Therefore, we will rank the ligands based on their pearson correlation coefficient.

```
ligand_activities %>%
    arrange(-pearson)
## # A tibble: 89 x 4
##    test_ligand auroc   aupr pearson
##    <chr>       <dbl>  <dbl>   <dbl>
##  1 IL23A       0.742 0.0693   0.173
##  2 TNF         0.753 0.0604   0.165
##  3 TNFSF13B    0.732 0.0568   0.159
##  4 IL1A        0.712 0.0532   0.155
##  5 LAMA2       0.740 0.0597   0.152
##  6 ICAM4       0.731 0.0645   0.151
##  7 L1CAM       0.735 0.0645   0.151
##  8 CXCL9       0.742 0.0771   0.151
##  9 NPPB        0.724 0.0721   0.151
## 10 INHBA       0.677 0.0591   0.150
## # ... with 79 more rows
best_upstream_ligands <- ligand_activities %>%
    top_n(12, pearson) %>%
    arrange(-pearson) %>%
    pull(test_ligand)
head(best_upstream_ligands)
## [1] "IL23A"    "TNF"      "TNFSF13B" "IL1A"     "LAMA2"    "ICAM4"
```

For the top 12 ligands, we will now build a multi-ligand model that uses all top-ranked ligands to predict whether a gene belongs to the inflammatory response program of not. This classification model will be trained via cross-validation and returns a probability for every gene.

```
## To increase these numbers.
k = 3 # 3-fold
n = 10 # 10 rounds
inflammation_gene_predictions_top12_list <- seq(n) %>%
    lapply(assess_rf_class_probabilities,
           folds = k,
           geneset = geneset_oi,
           background_expressed_genes = background_expressed_genes,
           ligands_oi = best_upstream_ligands,
           ligand_target_matrix = ligand_target_matrix)
```

Evaluate now how well the target gene probabilies agree with the gene set assignments

```
# get performance: auroc-aupr-pearson
target_prediction_performances_cv <-
    inflammation_gene_predictions_top12_list %>%
    lapply(classification_evaluation_continuous_pred_wrapper) %>%
```

```
    bind_rows() %>%
    mutate(round=seq(1:nrow(.)))
```

We display here the AUROC, AUPR and PCC of this model (averaged over cross-validation rounds)

```
target_prediction_performances_cv$auroc %>% mean()
## [1] 0.70906
target_prediction_performances_cv$aupr %>% mean()
## [1] 0.0306664
target_prediction_performances_cv$pearson %>% mean()
## [1] 0.09457835
```

Evaluate now whether genes belonging to the gene set are more likely to be top-predicted. We look at the top 5% of predicted targets here.

```
## get performance: how many inflammatory genes and inflammatory genes among
## top 5% predicted targets
target_prediction_performances_discrete_cv <-
    inflammation_gene_predictions_top12_list %>%
    lapply(calculate_fraction_top_predicted, quantile_cutoff = 0.95) %>%
    bind_rows() %>%
    ungroup() %>%
    mutate(round=rep(1:length(inflammation_gene_predictions_top12_list), each = 2))
```

What is the fraction of inflammation related genes that belongs to the top 5% predicted targets?

```
target_prediction_performances_discrete_cv %>%
    filter(true_target) %>%
    .$fraction_positive_predicted %>%
    mean()
## [1] 0.3016129
```

What is the fraction of non-inflammation related genes that belongs to the top 5% predicted targets?

```
target_prediction_performances_discrete_cv %>%
    filter(!true_target) %>%
    .$fraction_positive_predicted %>%
    mean()
## [1] 0.0496038
```

We see that the inflammation genes are enriched in the top-predicted target genes. To test this, we will now apply a Fisher's exact test for every cross-validation round and report the average p-value.

```
target_prediction_performances_discrete_fisher <-
    inflammation_gene_predictions_top12_list %>%
    lapply(calculate_fraction_top_predicted_fisher, quantile_cutoff = 0.95)

target_prediction_performances_discrete_fisher %>% unlist() %>% mean()
## [1] 3.250723e-08
```

Finally, we will look at which genes are well-predicted in every cross-validation round.

```
# get top predicted genes
top_predicted_genes <- seq(length(inflammation_gene_predictions_top12_list)) %>%
    lapply(get_top_predicted_genes,inflammation_gene_predictions_top12_list) %>%
    purrr::reduce(full_join, by = c("gene","true_target"))
top_predicted_genes %>% filter(true_target)
## # A tibble: 27 x 12
```

```
##    gene  true_target predicted_top_t... predicted_top_t... predicted_top_t...
##    <chr> <lgl>       <lgl>             <lgl>             <lgl>
##  1 NFKB... TRUE        TRUE              TRUE              TRUE
##  2 SELE  TRUE        TRUE              TRUE              TRUE
##  3 IRF1  TRUE        TRUE              TRUE              TRUE
##  4 NFKB1 TRUE        TRUE              TRUE              TRUE
##  5 EIF2... TRUE        TRUE              TRUE              TRUE
##  6 MYC   TRUE        TRUE              TRUE              TRUE
##  7 RIPK2 TRUE        TRUE              TRUE              NA
##  8 LYN   TRUE        TRUE              NA                TRUE
##  9 CDKN... TRUE        TRUE              TRUE              TRUE
## 10 PTAFR TRUE        TRUE              TRUE              TRUE
## # ... with 17 more rows, and 7 more variables: predicted_top_target_round4 <lgl>,
## #   predicted_top_target_round5 <lgl>, predicted_top_target_round6 <lgl>,
## #   predicted_top_target_round7 <lgl>, predicted_top_target_round8 <lgl>,
## #   predicted_top_target_round9 <lgl>, predicted_top_target_round10 <lgl>
```

References

Browaeys, R., Saelens, W. & Saeys, Y. NicheNet: modeling intercellular communication by linking ligands to target genes. Nat Methods (2019) doi:10.1038/s41592-019-0667-5

Puram, Sidharth V., Itay Tirosh, Anuraag S. Parikh, Anoop P. Patel, Keren Yizhak, Shawn Gillespie, Christopher Rodman, et al. 2017. "Single-Cell Transcriptomic Analysis of Primary and Metastatic Tumor Ecosystems in Head and Neck Cancer." Cell 171 (7): 1611–1624.e24. https://doi.org/10.1016/j.cell.2017.10.044.