artdeco vignette

Raik Otto

2018-12-14

artdeco R package

A R package that determine the differentiation stage of pancreatic neuroendocrinal neoplasia (PANnen) based on bulk RNA-seq of or mRNA array exepriments on cancer-tissue.

1 How to make it work: Quickstart

Installing artdeco

Start an R session e.g. using RStudio

```
if (!requireNamespace("BiocManager", quietly=TRUE)) install.packages("BiocManager")
```

```
BiocManager::install("artdeco")
```

Test run

57 representative PANnen bulk RNA-seq samples are shipped along with the R package. These will in the following be analyzed by the artdeco algorithm to demonstrate the utilization of artdeco.

```
library("artdeco")

# obtain path to testdata
path_transcriptome_file = system.file(
    "/Data/Expression_data/PANnen_Test_Data.tsv",
    package="artdeco")

# testrun
deconvolution_results = Determine_differentiation_stage(
    transcriptome_file_path = path_transcriptome_file
)
```

```
## [1] "No meta data sheet provided, creating meta data sheet from transcriptome file."
## [1] "Model(s) loaded"
## [1] "Deconvolving with model: Alpha_Beta_Gamma_Delta_Lawlor"
```

```
# show results
deconvolution_results[1:5,1:5]
```

```
##    Sample beta_similarity beta_similarity_percent alpha_similarity
## S1     S1            none                       8             none
## S2     S2            none                       7             none
## S3     S3          traces                      12             none
## S4     S4            none                      10             none
## S5     S5            none                      10             none
##    alpha_similarity_percent
```

```
## S1                          0
## S2                          0
## S3                          0
## S4                          0
## S5                          0
```

The call returns a dataframe that contains the differentiation stage similarity predictions.

Explanation test run results

Percent-wise similarities of the testdata samples to all scRNA derived training samples are shown, both as written interpretation e.g. “significant similarity (to this cell type)” or “no significant similarity (to this cell type)” as well as percent scalars.

Explanation percent similarities

The percent value quantifies to what extend a given transkriptome can be reconstructed utilizing a given reference cell type e.g. by the insulin producing ‘beta’ cell type as basis. The nu-SVM regression based similarity determination has the limitation that the returned unit-less scalar is challenging to interpret: It is not clearly decidable when a scaler indicates a high or low similarity. Thus, artdeco interprets the returned scalar. Measurements may be calculated based on a trained model (absolute) or relative to all analyzed of within a given run (relative). The default ‘absolute’ mode interprets the return scalar for a given samples relative to the maximal similarity measured for each model subtype during training. I.e. the returned scaler is divided by the maximal similarity observed for training set samples. E.g. when a scRNA beta cell received the similarity 5.7 to the beta set, than all subsequent beta similarities are divided by 5.7 to show how similar they are relative to the self-identification of cell subtypes. The ‘relative’ mode divides by the maximal similarity measured for a given set of query data, i.e. the maximally measured similarity is taken as divisor.

2 Visualization of similarity predictions

It is important to interpret the simialrity predictions given the clustering of the data. In order to detect clusters, a correlation heatmap can be used along with an overlay of similarity predictions. The rational is, that samples that show a comparable differentiation state should cluster together, which is why a heatmap creator is included in the artdeco package and can be utilized as follows. First the analyzed transcriptome data is load and afterwards the similarity predictions from step one are being passed on to the visualization function.

```
visualization_data_path = system.file(
    "/Data/Expression_data/Visualization_PANnen.tsv",
    package="artdeco")

create_heatmap_differentiation_stages(
    visualization_data_path,
    deconvolution_results
)
```

3 Adding and removing models

Adding a new model requires 1) training data and 2) a labeling vector of the training data specifying the cell type of each training sample

```
meta_data_path = system.file("Data/Meta_Data.tsv", package = "artdeco")
meta_data      = read.table(
    meta_data_path, sep ="\t", header = T,
    stringsAsFactors = F)
subtype_vector = meta_data$Subtype # extract the training sample subtype labels

subtype_vector[1:6] # show subtype definition
```

```
## [1] "Alpha" "Alpha" "Alpha" "Alpha" "Alpha" "Alpha"
```

```
training_data_path = system.file(
    "Data/Expression_data/PANnen_Test_Data.tsv", package = "artdeco")

add_deconvolution_training_model(
    training_data = training_data_path,
    model_name = "My_model",
    subtype_vector = subtype_vector,
    training_nr_marker_genes = 5
)
```

```
## [1] "Loading training data"
## [1] "Calculating marker genes for subtype: alpha"
## [1] "Calculating marker genes for subtype: beta"
## [1] "Calculating marker genes for subtype: gamma"
## [1] "Calculating marker genes for subtype: delta"
## [1] "Calculating marker genes for subtype: acinar"
## [1] "Calculating marker genes for subtype: ductal"
## [1] "Finished extracting marker genes for subtypes"
## [1] "Basis trained, estimating deconvolution thresholds,this may take some time"
## [1] "Finished threshold determination"
## [1] "Storing model: /home/ottoraik/R/x86_64-pc-linux-gnu-library/3.5/artdeco/Models//My_model.RDS"
## [1] "Finished training model: My_model"
```

Note that the training in the current version (1.0.0) of artdeco has to have HGNC symbols as rownames.

Remove models is fairly straight forward:

```
models = list.files(system.file("Models/", package = "artdeco"))
print(models)
```

```
## [1] "Alpha_Beta_Gamma_Delta_Acinar_Ductal_Lawlor.RDS"
## [2] "Alpha_Beta_Gamma_Delta_Baron_Hisc_Haber.RDS"
## [3] "Alpha_Beta_Gamma_Delta_Baron_Progenitor_Stanescu_HESC_Yan.RDS"
## [4] "Alpha_Beta_Gamma_Delta_Baron_Progenitor_Stanescu.RDS"
## [5] "Alpha_Beta_Gamma_Delta_Baron_Progrenitor_Stanescu_Hisc_Haber.RDS"
## [6] "Alpha_Beta_Gamma_Delta_Lawlor.RDS"
## [7] "Alpha_Beta_Gamma_Delta_Segerstolpe_Progenitor_Stanescu_Hisc_Haber.RDS"
## [8] "My_model.RDS"
```

```
model_to_be_removed = models[length(models)]
remove_model(
    model_name = model_to_be_removed
)
```

```
## [1] "Deleted model: My_model"
```

4 Important: data format

Please note that the expression data always has to have HGNC symbols as rownames and that the first entry of the column names indicates the hgnc symbols, not the first sample name.

```
training_data_path = system.file(
    "Data/Expression_data/PANnen_Test_Data.tsv", package = "artdeco")
expression_matrix = read.table(
    training_data_path,
    sep ="\t",
    header = T,
    row.names = 1)

expression_matrix[1:5,1:5]
```

```
##                S1           S2           S3           S4           S5
## INS  1.287430e+02   402.175244    0.0354701 1.658323e+01 4.805062e-01
## COX1 1.426009e+04 20922.117339 9829.7618331 1.768796e+04 1.295777e+04
## IAPP 5.249958e-01     3.213968    0.1044866 1.627951e-02 7.418050e-03
## ND4  4.721256e+03  7036.506560 2600.5447703 1.804696e+03 3.997664e+03
## CYTB 4.526890e+03  5329.243390 1262.9641899 3.803131e+03 2.531839e+03
```

Contact: raik.otto@hu-berlin.de