Redeconve manual

```
#library(Redeconve)
```

Main function

This part describes how to use the main function `deconvoluting` to perform single-cell deconvolution.

Useage

The usage of `deconvoluting` is as follows:

```
## nums = deconvoluting(ref, st, cellnames=NULL, genemode, gene.list, var_thresh=0.025,
         exp_thresh=0.03, hpmode, hp, aver_cell, thre=1e-10, dopar=T, ncores, realtime=F,
         dir=NULL)
```

It does contains many parameters. Next, I will divide these parameters into several parts by function, and explain them one by one.

1. Necessary data for deconvolution

`ref` and `st` are the core data used for deconvolution. They are required as follows:

- `ref`, the scRNA-seq data served as reference for deconvolution. It is a `Matrix` (or `dgCMatrix`) of unprocessed count-level scRNA-seq data. One row represents one gene and one column represents one cell.
- `st`, the spatial transcriptomics to be deconvoluted. It is a `Matrix` (or `dgCMatrix`) of unprocessed spatial transcriptomics, represents raw counts in each spot. One row represents one gene and one column represents one spot.

2. Mode of gene selection

`genemode`, `gene.list`, `var_thresh` and `exp_thresh` are about how to deal with genes. `genemode` determines the mode of handling genes and `gene.list`, `var_thresh`, `exp_thresh` are associated with specific modes. Redeconve offers 3 alternative modes of dealing with genes:

1. `default`: Use the intersection of genes in `ref` and `st`, without other treatment.
2. `customized`: Indicating gene list yourself. Parameter `gene.list` is the list of genes you indicated. Note that only those genes within the intersection of `ref` and `st` would be used.
3. `filtered`: We will use a built-in function `gene.filter` to screen some genes. This function will first take the intersection of `ref` and `st`, the use two indices, `var_thresh` and `exp_thresh` to filter genes. You can customize these two parameters as well.

- `var_thresh` considers variance of reference. Genes whose variance across all cells in reference do not reach that threshold will be filtered out. The default value is 0.025.
- `exp_thresh` considers expression in spatial transcriptomics. Genes whose average count across all spots in spatial transcriptomics is less that this value will be filtered out. The default value is

0.003.

3. Mode of determining hyperparameter

The hyperparamter is our key to single-cell resolution (See Methods for details). Here we still offers 3 modes to determine the hyperparameter:

1. `default`: We will calculate a hyperparameter according to the number of genes and cells in reference (See Methods for details).
2. `customized`: Indicating the hyperparameter yourself.
3. `autoselection`: Redeconve will use a procedure to select the optimal hyperparameter. In this procedure, a series of hyperparameter will be set in the vicinity of the hyperparameter selected by mode `default`, and Redeconve will use these hyperparameters to perform deconvolution separately, then return the result with the best hyperparameter. You can see Methods for details about how we determine the best hyperparameter.

- Note that in this procedure, several rounds of deconvolution will be performed, so it may take a long time. Under such circumstances, parallel computing will be beneficial.

4. Parallel computing

Sometimes the reference will contain tens of thousands of cells, or the spatial transcriptomics will contain tens of thousands of spots (e.g. when the data is from Slide-seq), then parallel computing is useful. Redeconve uses the package “doSNOW” to achieve parallel computing (which means there is a progress bar). Related parameters are `dopar` and `ncores`.

- `dopar` determines whether to use parallel computing or not.
- `ncores` indicates the number of cores to be used in parallel computing. It’s recommended to manually set this parameter rather than use the function `detectCores` to avoid underlying errors.

! Important tips for parallel computing: !

1. Our underlying algorithm makes use of OpenBLAS, which may include parallel computing inside. Therefore, setting `system("export OPENBLAS_NUM_THREADS=1")` is necessary to avoid underlying errors.
2. An error may be reported when the number of threads is too large : `Error in socketAccept(socket = socket, blocking = TRUE, open = "a+b",: all connections are in use`. If such error occurs, please reduce the number of cores.

5. Writing real-time results

Even with parallel computing, some dataset is still time-consuming. Redeconve is able to write results into disk in real time at the cost of some running speed. Related parameters are `realtime` and `dir`.

- `dopar` determines whethers to write the results into disk in real time or not.
- `dir` indicates the directory to write the results.

For real time results, the result of each spot will be write into a separate csv file, whose name is the barcode of the spot.

6. Other parameters

The left parameters are `cellnames`, `normalize` and `thre`.

- `cellnames`: Chances are that you may not want to use all cells in reference to run deconvolution. Then you can indicate which cells will be used by this parameter. If you do not specify this parameter, all cells will be used.
- `normalize`: Redeconve can also be used for bulk RNA-seq deconvolution. When doing this, normalization for reference is not required. When deconvoluting spatial transcriptomics, normalization is recommended.
- `thre`: The estimated cell abundance will not be exactly 0. This parameter indicates that the abundance less than this value will be treated as 0. Generally this value does not need to be adjusted, and the result will remain the same within a relatively big range of this value.

A demo

Next we will use a demo to give an example of how to use this function.

```
## load the data
#data(basic)

## check the dimensions of sc and st
#dim(sc)
#dim(st)

## check the number cells in each cell type
#table(annotations[,2])

## deconvolution
# this may take a long time
#res = deconvoluting(sc,st,genemode="filt",hpmode="def",aver_cell=25,dopar=T,ncores=8)
```

- `sc` and `st` are separately reference and spatial transcriptomics.
- For there are about 20000 genes, `genemode` is set to `"filtered"` with the default threshold of variance and mean expression, which result in #### genes.
- `hpmode` is set to `"default"` to improve efficiency.
- `dopar` is set to `TRUE` (default value) and `ncores` is set to 8. You can raise the number of cores to improve efficiency.
- This dataset is from the ST platform whose spot radius is about 100 μm, so we set `aver_cell` as 25.
- For this dataset is not very large, `realtime` is set to `FALSE` (default value).
- We want to use all cells in deconvolution, so we do not need to specify `cellnames`. Also, we do not need to adjust `thre`.

Cell-type deconvolution

Like other methods, Redeconve can also perform deconvolution at cell-type level. This part shows how to do so.

```
## get reference
#ref = get.ref(sc,annotations)

## deconvolution
#res.ct = deconvoluting(sc,st,genemode="filt",hpmode="auto",dopar=T,ncores=8)
```

You can see that actually, only one more step is required to convert single-cell expression profile to that of cell type. The function `get.ref` will take the average expression of all cells in one cell type as the profile of that cell type.

For the main function, every thing is the same. Here we set `hpmode` as `"autoselection"`, for there are only tens of cell types, then the speed is fast enough for us to run several rounds of deconvolution.

Seurat interface

This package does not directly provide Seurat interface. However, The input we need can be easily extracted from Seurat object:

```
# reference: raw count
# pbmc: Seurat object example
#sc = pbmc@assays$RNA@counts

# spatial: count and coordinates
# stxBrain: Seurat object example
#st = stxBrain@assays$spatial@counts
#coords = test@images$anterior1@coordinates[,c("row","col")]
```

Downstream analysis and visualization

Redeconve offers many built-in functions for downstream analysis and visualization.

gaining interpretability

For we included a normalization procedure, the result of `deconvoluting` has only relative significance. Here are some functions related:

1. `to.proportion`: This function converts the result to proportion, i.e., the sum of cell abundance per spot is 1. This is convenient for visualization.

```
#prop = to.proportion(nums)
```

2. `to.absolute.abundance`: This function converts the result to absolute cell abundance with some priori knowledge. Parameter `aver.cell` is required to estimate the absolute abundance of each cell state. This value indicates the average number of cells in one spot. Users can determine it according to the platform of st data. For example, 10x Visium has a spot radius of about 50μm, so this value for a 10x Visium dataset is about 10.

```
#ab = to.absolute.abundance(nums,aver.cell=10)
```

3. `sc2type`: This function converts single-cell result to cell-type result. Note that this function itself does not provide interpretability.

```
#type_res = sc2type(res,annotations)
```

Cell occurrence

The function `cell.occur` shows the number of cells actually used in deconvolution, and the number of spots that every used cell occurs.

```
#cell.occur(res)
```

Find cells of interest

Redeconve offers a function to highlight those cells with high average abundance and/or high coefficient of variation (sd/mean). Users can then explore those cells. Note that this function can only be run after `cell.occur`, because it needs `occurred.cells` as input, which is an output of `cell.occur`.

```
#show.cellsofinterest(res, occurred.cells)
```

Abundance of all or some specific cells

The function `cell.type.weight` can plot the abundance of all cells (or cell types) or some cells you indicate. Here we want to plot the abundance of three T cells, so we set `cell.type = F` and `cellnames` as the names of cells we concern.

```
#Tcells = c("T.cells...NK.cells.8","T.cells...NK.cells.11","T.cells...NK.cells.35")
#cell.type.weight(res,F,cellnames=Tcells,coords=coords,name="Tcells.pdf")
```

The output file, named “Tcells.pdf”, can be found under current directory.

Spatial pie chart

The function `spatial.pie` can plot the spatial pie chart, showing the proportion of each cell type in each spatial spot.

```
# For this is a demo, we use default colors
# This may take some time
#spatial.piechart(res.ctmerge,coords)
```

Spot pie chart

The function `spot.pie` can plot the proportion of each cell type within a certain spot.

```
#spot.pie(res[,"X27x18"],title="X27x18")
```

Co-localization

Redeconve offers two main functions to explore co-localization. `coloc.corr` uses the correlation of cell abundance to infer co-localization, and `coloc.network` gains a network to visualize co-localization.

```
# This function merges single cells to cell types
#res.ctmerge = sc2type(res,annotations)

#corr = coloc.corr(res.ctmerge,F,method="pearson")

#celltypes = levels(as.factor(annotations[,2]))
#g = coloc.network(corr,thre=0.4,cell.type=T,annotations=celltypes,ntypes=length(celltypes))
#plot(g)
```

Spatial expression visualization

The function `spatial.gene` can visualize gene expression across spatial coordinates. This can help validate the distribution of some cells/cell states.

```
#gene.list = c("CD3D","CD3E","CD3G")
#spatial.gene(st,coords,gene.list)
```

Spatial expression profile imputation and reference correction

In spatial transcriptomics, drop-out is very severe. Meanwhile, the cell state of cells in spatial spot would not be exactly the same with that in reference. Here Redeconve offers a function that solves both problems. The function `profile.correction` can do impitation as well as profile correction for a single spot.

```
#ests = profile_correction(res[,"X27x18"],st[,"X27x18"],sc,ncores=8)
```

This is another quadratic programming problem: the goal is to let the corrected spatial expression higher than the observed (imputation). The return value is the “real” cell states in that spot after imputation.

We can use the function `profile_comparison` to see the difference between reference and corrected expression profile:

```
#profile_comparison(sc,ests)
```