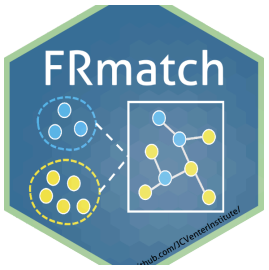
Oct 02, 2020

FR-Match: cell type matching for scRNAseq data



In 1 collection

DOI
https://dx.doi.org/10.17504/protocols.io.bmyfk7tn

Yun Renee Zhang[1], Brian Aevermann[1], Richard Scheuermann[1]

[1]J. Craig Venter Institute

Human Cell Atlas Metho...

Yun Renee Zhang
J. Craig Venter Institute

OPEN ACCESS

DOI: https://dx.doi.org/10.17504/protocols.io.bmyfk7tn

External link: https://github.com/JCVenterInstitute/FRmatch

Protocol Citation: Yun Renee Zhang, Brian Aevermann, Richard Scheuermann 2020. FR-Match: cell type matching for scRNAseq data. **protocols.io** https://dx.doi.org/10.17504/protocols.io.bmyfk7tn

Manuscript citation:
https://doi.org/10.1101/2020.05.01.073445

Protocol status: Working
We use this protocol and it's working

Created: October 02, 2020

Last Modified: October 02, 2020

Protocol Integer ID: 42727

Keywords: single cell RNA sequencing, cell types, data integration, cell type matching for scrnaseq data fr, cluster cell transcriptome integration across scrnaseq experiment, scrnaseq data fr, supervised cell phenotype matching strategy for cluster, supervised cell phenotype matching strategy, cell type matching, transcriptome integration, scrnaseq experiment, cluster cell, cluster, match,

Abstract

FR-Match is a supervised cell phenotype matching strategy for cluster-to-cluster cell transcriptome integration across scRNAseq experiments.

An R package and Shiny application are provided at https://github.com/JCVenterInstitute/FRmatch.

Before start

Software

R programming language	NAME
The R Foundation	DEVELOPER
Comprehensive R Archive Network	REPOSITORY
https://cran.r-project.org/	SOURCE LINK

Require R Shiny package.

Launch Shiny app

1 Interactively explore and match scRNAseq cell type clusters with the seamless Shiny app. The Shiny app may serve as a quick start demo with pre-loaded datasets.

Command

new command name

```
runShiny()
```

Data preparation and exploration

2 Use the built-in data preparation function to create data objects with required and optional input data elements. Create data objects for experiment 1 (E1) and experiment 2 (E2)

Command

new command name

```
dat_E1 <- make_data_object()
dat_E2 <- make_data_object()
```

2.1 View comparative cell cluster sizes.

Command

new command name

```
plot_clusterSize(dat_E1, dat_E2)
```

2.2 View "barcode" plot for cluster of interest.

Command

new command name

```
plot_cluster_by_markers(dat_E1, cluster.name = "cluster_of_interest")
```

Run main algorithm

10m

3 Use wrapper function to perform bi-directional matching.

3.1 Map E1 to E2.

Command

new command name

```
rst12 <- FRmatch(sce.query = dat_E1, sce.ref = dat_E2)
```

3.2 Map E2 to E1.

Command

new command name

```
rst21 <- FRmatch(sce.query = dat_E2, sce.ref = dat_E1)
```

Combine and plot matching results

4 Combine the bi-directional matching results and plot.

Command

new command name

```
plot_bi_FRmatch(rst12, rst21)
```

Additional plots

5 Some optional plotting functions to help studying the matching results.

5.1 Plot one-directional matching results.

Command

new command name

```
plot_FRmatch(rst12)
```

Plot one-directional matching p-values.

Command

new command name

```
plot_FRmatch(rst12, type = "padj")
```

5.2 Minimum spanning tree (MST) plot. MST can be plotted by turning on the plot option in the test function.

Command

new command name

```
FR.test(samp1, samp2, plot.MST = TRUE)
```