Analysis Readme

Project description: scRNA-seq of mouse intenstine organoids co-cultured with Tregs and/or cytokines/cytokine receptor inhibitors.

This Readme explains how to work with the analysis code of this project.

Preparing data

The Cellranger input count data is not part of this repository and must be gathered manually from GEO (accession number provided in manuscript). It must then be placed in *experiments/scRNA/analysis/cellranger_20230419/[Library name]/outs/filtered_feature_bc_matrix*, as files named *barcodes.tsv.gz*, *features.tsv.gz* and *matrix.mtx.gz*, according to the type of data. The files currently existing in these directories are dummies. The Haber et al. (2017) reference data (GEO accession number: GSE92332) used for cell annotation must be placed in *experiments/scRNA/reference_data*, with the name provided by the dummy file located there.

Folder structure

The analysis can be found in the folder *experiments/scRNA/analysis/custom/hep62703_2023-05-04_B_harmony*.

- scripts: R and Rmd analysis scripts
- fig: Figures/plots
- data: Data saved during analysis, e.g. Seurat objects at different stages
- table: Result tables
- metadata: Metadata related to this analysis

Running scripts

Scripts are run with the *R* (or *Python*) version and package versions specified in the manuscript methods, and use the files *config.yaml*, *analparams.yaml*, *analparams_preprocessing_[DATE].yaml* and/or *analparams_preprocessing_filters_[DATE].yaml* for retrieving parameters. In *config.yaml*, the path to the Cellranger input count data, as well as the analysis output path and the species are defined. The analparams files are used for setting the other parameters. The analparams filters file additionally contains the criteria for cell filtering. The paths to the config/analparams files are specified at the beginning of each script. When re-running the pipeline, the *analparams_preprocessing_[DATE].yaml* will be created newly with the **Preprocessing.Rmd** script, with the current date. The *analparams_preprocessing_filters_[DATE].yaml* file has to be created manually by adding cell filtering criteria to the *analparams_preprocessing_[DATE].yaml* file. How these filters are set can be seen in the *analparams_preprocessing_filters_[DATE].yaml* file already existing in the repository as an example. The *analparams_preprocessing_filters_[DATE].yaml* then has to be used by **QC_report_filters.Rmd** and all subsequent scripts.

All paths in the scripts are specified relative to the project root folder, and the project root folder is defined via the dummy file .projroot, using the rprojroot package. Nevertheless, *R* should generally be started in the project root folder. Not all Rmd-files are actually run as reports, some are just run as simple scripts to generate data. Saving data may be disabled by a variable (e.g., *saveData*) at the beginning of the script. If the script should automatically save its output data when sourcing/rendering it, this variable has to be set to *TRUE*. When re-running the pipeline, the name

of the input data loaded at the beginning has to be changed in many of the scripts, as data is often saved with the date of creation in the name.

Scripts in the order in which they should be run

- **Preprocessing.Rmd**: Preprocessing of Cellranger count data (gene expression, HTOs) to obtain a Seurat object with HTO-based assignment of groups,
  creating a report with diagnostic plots mainly on HTO labeling
- **QC_report.Rmd**: QC report on cell quality metrics, unfiltered cells
- **QC_report_filters.Rmd**: QC report on cell quality metrics, with filters, saves Seurat object with filter criteria (but does not filter it yet!)
- **find_doublets_scDblFinder_cluster-no-neg.R**: Finds doublets using scDblFinder, saves scDBlFinder results
- **scDblFinder_analysis_cluster-no-neg.Rmd**: Report on scDblFinder results
- **set_doublets_scDblFinder.R**: Sets doublets from scDBlFinder results in Seurat object and saves Seurat object (must be run after QC report)
- **clustering.R**: Count data preprocessing, dimensionality reduction, harmony integration, UMAP and clustering. Saves filtered and clustered Seurat object
- **clustering_analysis.Rmd**: Report on clustering results
- **annotation_SingleR_res_1.R**: Script to run SingleR-based annotation. Saves SingleR results for single cell and cluster based annotation, as well as Seurat object with SingleR annotation in metadata. Annotation based on Haber et al. (2017)
- **annot_SingleR_analysis_res_1.Rmd**: Report on SingleR annotation results
- **marker_scores.Rmd**: Computation of Module and UCell scores of marker genes. Saves Seurat object with scores
- **annot_analysis.Rmd**: Report with comprehensive annotation analysis including UMAP plots of clustering and annotation, Module and UCell scores, marker gene heatmaps
- **diff_expr_conditions_nebula.Rmd**: Fit nebula model for differential gene expression analysis between conditions for each celltype. Saves nebula model
- **comparison_statistics_conditions_nebula.Rmd**: Extract statistics for desired comparisons from nebula model using contrasts, and save results
- **gsea_conditions_nebula.Rmd**: Pathway analysis. Compute GSEA statistics on nebula results and save GSEA results
- **gsea_analysis_conditions_nebula.Rmd**: Report on GSEA results, shows significant results at FDR 10% as tables for each celltype and comparison
- **pathway_analysis_heatmaps_cond.Rmd**: Report that creates (dot) heatmaps from GSEA pathway results using a defined selection of pathways from csv
- **pathway_analysis_venn_cond.Rmd**: Report that creates Venn diagrams of pathway results with the ggvenn package. Also computes pathway
  correlations between comparisons and plots them. Saves pathway set data that can be used as input for creating Venn diagrams as yaml file. The Venn diagrams for the manuscript are not created with this this script/ggvenn, but in Python with matplotlib-venn
- **pathway_analysis_venn_cond_mpl.ipynb**: Jupyter notebook that uses **Python and matplotlib-venn** to create Venn diagrams scaled by set size

- **get_cell_counts.R**: Script that generates a cell count matrix from the scRNA-seq cell type annotation, and saves that matrix