Raredon Laboratory

Single Cell Analysis Workflow

Version 1.0

2024-03-11

MSBR

Step 0: Get access to the right hardware and software

Before beginning this workflow:

1. Look up your computer specifications to see if it can handle this workflow locally. Many of these tasks can be done on a home computer. Some are resource-intensive and require high-performance hardware.
2. Get access to the clusters and familiarize yourself with how they work. Practice moving files between your lab hard-drive and the Yale HPCs with Globus. Learn the Yale HPC file structure and where you are and are not allowed to store large files (read the information online, it changes regularly).
3. I recommend at least four pieces of software be installed on your local device:
   1. R
   2. R Studio
   3. Globus Connect
   4. GitHub Desktop
4. Identify GitHub repositories to clone to your local machine
   1. Learn how to pull and push commits using GitHub desktop
   2. Build yourself a project-specific GitHub ‘sandbox’ to work in – a file folder where you can do anything you want and will be backed up to the cloud

Legend:

**Master Objects (Saved Permanently)**

**Reference Objects (Saved Temporarily)**

*~ Required input objects*

* *description of output*

Step 1: Getting to a Clean Data Object

*~Requires FASTQ files as input*

* + *yields cleaned, classed, and annotated objects as output*
* Sequencing
  + yields
    - * **FASTQ** **files** (from YCGA or external sequencing company)
* Alignment
  + yields
    - * **Digital Gene Expression Matrix (DGEs**), Unfiltered
      * can also be made to output additional spliced and unsliced matrices, for RNA velocity calculations
* Import to R
  + yields
    - * Seurat Object, Unfiltered
* Filtration (conservative threshold-based filtration)
  + yields
    - * Seurat Object, Filtered (designed to be >25% ‘garbage’)
* Cleaning (clustering/embedding based cleaning approach)
  + yields
    - * Seurat Object, Clean (designed to be ~0-1% ‘garbage’)
* Classing
  + yields
    - * Seurat Object, Clean, Classed
* Primary Annotation (just to give us a first-pass idea of what we have captured
  + yields
    - * **Seurat Object, Clean, Classed, Annotated**
* Saving
  + yields
    - * A clean, classed, and annotated ".Robj" file for later loading and use

Step 2: Creating Merged and/or Integrated Phenotype Objects

*~Requires two or more clean, classed, annotated .Robj files as input*

* + *yields global phenotype objects as output*
* Merge
  + yields
    - * Seurat Object containing multiple Samples
* Integration
  + yields
    - * Seurat Object containing multiple Samples, re-structured in a way that is useful for cross-sample clustering and annotating
* Secondary Annotation (can include subcluster analysis)
  + yields
    - * **Global Phenotype Object**
* Save
  + yields
    - * A ‘complete’ phenotype object, ready for downstream analysis.

Step 3: Creating Phenotype Sub-Objects

~*Requires a complete phenotype object as input*

* + *yields organized sub-objects as output*
    - Certain projects may benefit from creating sub-objects in advance. These can be organized by class, timepoint, condition, etc.
    - For instance, if you are interested in studying global epithelial patterns in your project, it might be useful to create an epithelial object.
    - If it looks like a global integration obscures too much variance, but a sub-integration by experimental condition or timepoint seems useful, then you can make an object for each condition or timepoint.
    - Saving these objects in advance can save time later during Steps 6, 8, and 10.

Step 4: Creating NICHES Global Objects

~*Requires at least one complete phenotype object as input*

* Check Metadata
  + Consistent?
  + Complete?
  + Optimal for publication?
* Impute (resource intensive, clusters recommended)
  + Try to impute the entire global phenotype object, to preserve sample-sample variance trends of interest.
  + I recommend thresholding the features that ALRA considers, to only those expressed in at least 25-50 individual cells (this reduces downstream ‘false positives’ in NICHES analysis)
* Split Object By Sample
* Use SplitObject function in Seurat
* Check that the names of the list are the names of the Samples
* RunNICHES
* Run on both ‘RNA’ and ‘ALRA’ assay in parallel
* Be deliberate with each input argument
* Check that outputs make sense and look good
* Organize Outputs
  + yields
    - * **Global Connectomic Object**
* Save
  + yields
    - * A ‘complete’ connectomic object, ready for downstream analysis.

Step 5: Creating NICHES Sub-Objects

* NICHES objects are large and complex. I recommend breaking NICHES objects into smaller parts in advance. This can help us to better observe and explore patterns of interest.
* My recommendation is to break NICHES data *by receiving class* into multiple ‘class.receiving’ sub-objects. These usually are valuable and for biological exploration, produce interpretable results, and allow you to analyze an entire project using only 4-5 objects.
* Saving these objects in advance is useful for downstream Steps 7, 9 and 10.

Step 6: Statistical Analysis (Phenotype)

* This step involves designing a custom statistical testing workflow which yields a list of features which capture the ‘variance of interest’ in your dataset and can be studied in detail to reveal biologically-relevant findings.
* For all projects, it requires some experimentation and trial-and-error.
* Usually, a Wilcoxon Rank Sum test (within FindMarkers and FindAllMarkers in Seurat) is a sufficient base function. Our job is usually to figure out *how* to apply this function in a way that helps us compute the answers we want.
* List-wise operations can be very useful here, and allow intra-study meta-analysis of the resulting findings.
* Marker testing should be run on the raw ‘RNA’ assay whenever possible, and not on integrated or imputed ‘pseudo-value’ assays.

Step 7: Statistical Analysis (NICHES)

* The most powerful way to use NICHES is to define signaling archetypes. This lets the data structure itself guide the analysis.
* Then, we can ask how the cellular metadata aligns, or does not align, with the identified signaling archetypes.
* Careful differential analysis of NICHES data via carefully-chosen metadata handles can also yield meaningful results. However, I do not recommend doing this without first looking at signaling archetypes, which helps to inform us of the global NICHES data structure.

Step 8: Phenotype Visualizations

* Heatmaps are your friend. Learn how to use the ComplexHeatmap package in R.

<https://jokergoo.github.io/ComplexHeatmap-reference/book/>

* cowplot is helpful to compile complex multi-panel plots

<https://wilkelab.org/cowplot/articles/index.html>

* ggplot2 is useful for custom plot generation and Seurat plot fine-tuning

<https://rstudio.github.io/cheatsheets/data-visualization.pdf>

Step 9: NICHES Visualizations

* Everything you can do in phenotype land, you can do in NICHES land. But the metadata handling and interpretation is more complex, because each measurement has both sending and receiving metadata.
* This complexity is one of the reasons I recommending looking at one receiving class at a time, because results are easier to navigate.
* Any result identified in a class-wise object should be checked globally, in case you are ‘missing the forest for the trees’.

* I have built a couple of custom functions, currently in beta testing, to aid in the presentation of NICHES findings, including
  + CustomHeatmap
  + NetworkPlot
  + CircuitPlot

Step 10: Figure Craft

* Figures should be built in Adobe Illustrator (free via Yale Software Library). This is because Illustrator is a ‘vector’ program and is capable of producing infinitely-high resolution images for publication. It is the gold standard in graphic design. It is also a very powerful software tool that allows advanced figure craft.
* It is important that you familiarize yourself with how Illustrator works in advance of starting this step.
* The biggest thing to learn is that Illustrator references *local* *external files*, meaning that the user ‘points’ Illustrator to files on the local hard-drive but these files are not copied into the Illustrator document directly, they are simply ‘viewed’. This keeps figure file sizes small, and also allows rapid script-based updating of Illustrator figures in the background using R. It is a very powerful way to work.
* However, it comes with a caveat: Illustrator files will only build correctly if the ‘Links’ are correct. So, if you want to share a draft of your figure, you must ‘Export’ the Illustrator file to a .PNG or .PDF file for other users to view on other machines.
* Direct edits to Illustrator files should generally be performed by a single lead individual with control over all linked files.
* Broken links are a sign that Illustrator does not know where a required file is, and should be corrected before proceeding.
* Figures for publication should be .PNG files which are measured in real-world inches and exported at 300 dots-per-inch (dpi).

Step 11: Pre-print

* The first step in publishing is putting a ‘pre-print’ online on bioRxiv. This is the first major milestone which releases your work to the public. We will usually also Tweet out a narrative-style explanation of the work, to drum up press and to see who in the scientific community is most interested. This can help us to identify potential reviewers, and also fine-tune our selection of journal for submission.
* Usually we will want to submit to a journal within a month of posting the pre-print. This gives us a little time to choose a journal and polish things for submission.

Step 12: Peer Review

* This process can take time, and often goes beyond the length of people’s actual research appointments (I once had a paper published 7 years after I left an institution!)
* Because of this, it is important that lab members transfer their project knowledge so that the work can continue and our lab can respond to reviewer comments as a team.
* Before leaving, each person should convey to another, dedicated individual:
  + 1. How the existing files are organized
    2. Where the existing files are stored
    3. The meaning of important metadata annotations
    4. All scripts used to create the workflow to date
    5. Correlation with any physical samples, blocks, slides, etc. stored in lab