Code Documentation & Reproducibility

Nick Calistri

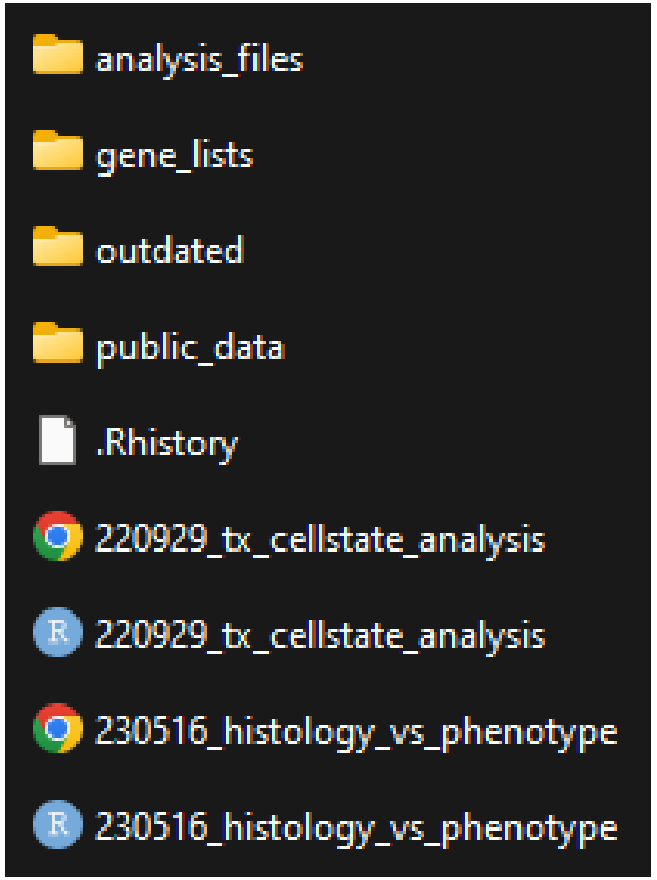
04/11/24

Sears Focus meeting

File management – keep it consistent!

- Filenames
  - Date_descriptor format automatically organizes within directory
    - 240411_sears_focus_code_organization.pptx

- Project directory structure
  - “/project_name/” – hosts code and report output
  - “/project_name/analysis_files/” – stores output of analysis
  - “/project_name/original_data/” – stores raw or original data
  - “/project_name/public_data/” – stores relevant public data
  - “/project_name/outdated/” – dumping ground for old versions of code that are no longer needed



Metadata organization

- Where feasible, keep metadata stored in .csv format to enable reproducible queries and programmatic access

Example 1: MycPten scRNA-seq metadata file

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	exa_dir	mpssr_seq_ru	mpssr_library_name	library_suffix	chemistry	reference_gen-t	cellranger_ver	feature_ba	condition_hto	condition_s	phenotype	misc_notes	histology
2	/home/gr	SCL210602RS	GEX1_Lib_XL	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-A0305_HTO_5	V3_T1	V		SP
3	/home/gr	SCL210602RS	GEX1_Lib_XL	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-A0306_HTO_6	V3_T2	V		SP
4	/home/gr	SCL210602RS	GEX1_Lib_XL	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-A0308_HTO_7	V4	V		SP
5	/home/gr	SCL210602RS	GEX1_Lib_XL	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-A0307_HTO_8	S3	S		SR
6	/home/gr	SCL210602RS	GEX2_Lib_XL	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-A0305_HTO_5	V3_T1	V		SP
7	/home/gr	SCL210602RS	GEX2_Lib_XL	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-A0306_HTO_6	V3_T2	V		SP
8	/home/gr	SCL210602RS	GEX2_Lib_XL	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-A0308_HTO_7	V4	V		SP
9	/home/gr	SCL210602RS	GEX2_Lib_XL	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-A0307_HTO_8	S3	S		SR
10	/home/gr	SCL210602RS	GEX1_Lib_CD	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-B0304_HTO_4	V5	V		SR
11	/home/gr	SCL210602RS	GEX1_Lib_CD	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-B0305_HTO_5	R2	R		SP
12	/home/gr	SCL210602RS	GEX1_Lib_CD	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-B0306_HTO_6	S4	S		SR
13	/home/gr	SCL210602RS	GEX2_Lib_CD	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-B0304_HTO_4	V5	V		SR
14	/home/gr	SCL210602RS	GEX2_Lib_CD	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-B0305_HTO_5	R2	R		SP
15	/home/gr	SCL210602RS	GEX2_Lib_CD	_reseq-hto	Single Cell 3' v3	mm10-2020-A	6.0.2	TRUE	TotalSeq-B0306_HTO_6	S4	S		SR

Metadata organization

- Where feasible, keep metadata stored in .csv format to enable reproducible queries and programmatic access

Example 2: MycPten cycIF metadata files

Sample metadata

Position II	Position	Core-ID	Tumor	slidescene	condition_string	treatment	scrna_string
A1	94	RS15-B	730L_COMBO_1	mTMA2-4_sceneA01	730L_COMBO_1	COMBO	PD_1
A4	83	RS10-A	573L_PARPi_R5_T2	mTMA2-4_sceneA04	573L_PARPi_R5_T2	PARPi_R	R5_T2
A8	86	RS11-B	772L_DT_1_R	mTMA2-4_sceneA08	772L_DT_1_R	PP2A	DT_1
A10	85	RS11-A	772L_DT_1_R	mTMA2-5_sceneA10	772L_DT_1_R	PP2A	DT_1
A12	93	RS15-A	730L_COMBO_1	mTMA2-5_sceneA12	730L_COMBO_1	COMBO	PD_1
B2	75	RS06-A	558L_PARPi_S5_T1	mTMA2-4_sceneB02	558L_PARPi_S5_T1	PARPi_S	S5_T1
B5	73	RS05-A	509L_PARPi_S2	mTMA2-4_sceneB05	509L_PARPi_S2	PARPi_S	S2
B9	95	RS16-A	729L_COMBO_2	mTMA2-4_sceneB09	729L_COMBO_2	COMBO	PD_2
C1	65	RS01-A	547L V4	mTMA2-4_sceneC01	547L V4	CONTROL	V4
C3	98	RS17-B	731L_COMBO_3	mTMA2-4_sceneC03	731L_COMBO_3	COMBO	PD_3

Feature metadata

column_name	data_type	marker_type	pass_qc	channel	round
subtractedregisteredimages	meta		TRUE	NA	NA
DAPI6	meta		TRUE	c1	6
FoxP3	quant	type	FALSE	c5	5
Ki67	quant	state	TRUE	c4	3
LamAC	quant	state	TRUE	c2	1
RAD51	quant	state	TRUE	c5	6
gH2AX	quant	state	TRUE	c4	6
pMYC	quant	state	FALSE	c4	2
pRPA	quant	state	TRUE	c3	2

Package/library management

- Virtual ‘environments’
  - Self-contained virtual workspaces
  - Enable reproducible computational results by securing the various packages or tools used for a given project
  - Safeguards issues arising from updating one package and it breaking your pipeline for another project
  - Typically helps with distribution of code – can share markup related to the environment so others can recreate it

- Recommendations:
  - Python : “Miniconda”
  - R : “renv”

Within your code

- Preamble
  - Relevant contact info and author identity
  - Brief description of code purpose

MycPten atlas:
s1_vehicle_stroma_preprocessing.html

Experiment & contact info

PIs: Rosalie Sears (searsr@ohsu.edu), Laura Heiser (heiserl@ohsu.edu)

Sample preparation: Zinab Doha (dohaz@ohsu.edu)

Library prep from single cells: Xi Li & Colin Daniel (danielc@ohsu.edu)

Analysis: Nick Calistri (calistri@ohsu.edu)

Sequencing performed by OHSU MPSSR

Analysis design

- Load each experiment individually
- Perform hashtag demultiplexing on each library individually
- Identify doublets with DoubletFinder
- Save so_list as .rds file

Within your code

- Preamble
  - Relevant contact info and author identity
  - Brief description of code purpose

- Set up
  - Load the libraries/packages needed to do the entire work
  - Load the data needed to do the entire work

MycPten atlas:
s1_vehicle_stroma_preprocessing.html

````
# Set up

## Load libraries
```{r}
library(Matrix)
library(tidyverse)
library(Seurat)
library(rliger)
library(SeuratWrappers)
library(ggalluvial)
library(DoubletFinder)
library(SoupX)
```

## Set a seed
```{r}
set.seed(1)
```

## Mouse/tumor stats from library_metadata file

```{r}
# Read in readme with HTO and tumor ID
lib_meta <- read_csv('library_metadata.csv')

# Add the run_library variable
lib_meta <- lib_meta %>%
  mutate(run_library = paste0(mpssr_seq_run_name,
                              '_',
                              mpssr_library_name,
                              str_replace_na(library_suffix, replacement = '')))

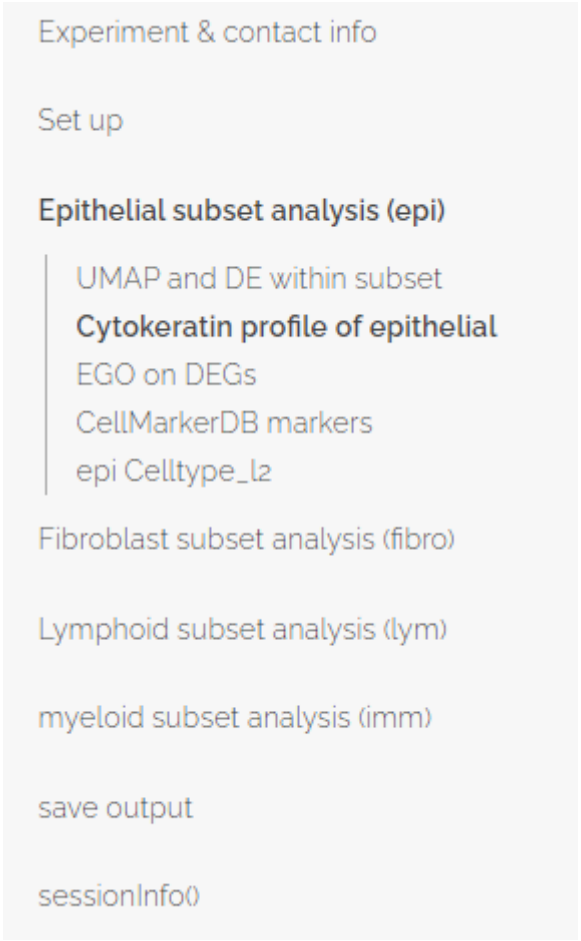
lib_meta$mouse_id <- str_split(lib_meta$condition_string, pattern = '_', simplify = TRUE)[,1]

lib_meta$phenotype <- str_replace(lib_meta$mouse_id, pattern = "[:digit:]+", "")
````
````

Within your code

- Preamble
  - Relevant contact info and author identity
  - Brief description of code purpose
- Set up
  - Load the libraries/packages needed to do the entire work
  - Load the data needed to do the entire work
- Core
  - Organize code chunks with descriptive headings
  - Nest sections where appropriate

MycPten atlas: s4_celltype_definition.html



Within your code

- Preamble
  - Relevant contact info and author identity
  - Brief description of code purpose
- Set up
  - Load the libraries/packages needed to do the entire work
  - Load the data needed to do the entire work
- Core
  - Organize code chunks with descriptive headings
  - Nest sections where appropriate
- End
  - Save output
  - R: run ‘sessionInfo’ to describe environment

MycPten atlas: s3_clustering_optimization.html

save rds output

```
saveRDS(so_merge, file = 'analysis_files/s3_celltypes.rds')
```

sessionInfo()

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] org.Mm.eg.db_3.13.0   AnnotationDbi_1.56.2  IRanges_2.28.0
##  [4] S4Vectors_0.32.3      Biobase_2.54.0        BiocGenerics_0.40.0
##  [7] clusterProfiler_4.0.5 harmony_0.1.0         Rcpp_1.0.7
## [10] bluster_1.2.1         cluster_2.1.2         SeuratObject_4.0.4
## [13] Seurat_4.1.0          forcats_0.5.1         stringr_1.4.0
## [16] dplyr_1.0.8           purrr_0.3.4           readr_2.1.2
## [19] tidyr_1.2.0           tibble_3.1.6          ggplot2_3.3.5
## [22] tidyverse_1.3.1       Matrix_1.3-4
##
```

Save computation time by automatically checking for code output!

```
if(file.exists('analysis_files/s2_vehicle_integrated.rds')){
  print('Loading existing s2_vehicle_integrated.rds file')
  so_merge <- readRDS('analysis_files/s2_vehicle_integrated.rds')
}else{
  print('so_merge_rliger.rds file not found.')
  print('Processing iNMF integration, 200k cells ~= 1 hour')
```

Simulated or subset test data

- Allows for verification that pipeline is working without large data transfer or computation time

Example: GSVA simulation

Simulate GSVA results

Just a random sampling from normal distribution

```
sim <- matrix(data = rnorm(n = length(hallmarks)*4),
              nrow = length(hallmarks)) %>%
  as_tibble(.) %>%
  mutate(gs = hallmarks)
```

Example: MycPten atlas

Full data:	3.60GB
Subset:	0.03GB

````
# Save output
```{r}
saveRDS(so_list,
        file = 'analysis_files/s1_seurat_list.rds')
```

# Save subset for pipeline demo
```{r}
n_cells <- 50

so_list_subset <- list()

for(i in 1:length(so_list)){
  cells_subset <- colnames(so_list[[i]])[sample(x = ncol(so_list[[i]]),
                                                size = n_cells,
                                                replace = FALSE)]

  so_list_subset[[i]] <- subset(so_list[[i]],
                                cells = cells_subset)
}

saveRDS(so_list_subset, file = 'analysis_files/s1_seurat_list_subset.rds')
```
````

Cleaning things out for publication

- Remove any unnecessary components*
- Add comments where appropriate
- Centralize figure generation to a final script
- Construct a .readme file that describes each included file
  - https://github.com/HeiserLab/NatureComms_MycPtenAtlas/tree/main

Miscellaneous coding best practices

- Keep thing simple, keep things interpretable
  - One line of code = do one single thing
- Avoid loops whenever possible
  - The “apply” class of functions can be orders of magnitude faster than a for loop

For loop example:

```
for(i in 1:length(a)){
  curr_a <- a[i]
  curr_b <- b[i]

  curr_product <- curr_a*curr_b
  curr_quotient <- curr_a/curr_b
  curr_exponent <- curr_a^curr_b
  curr_alogb <- log(curr_a, base = curr_b)

  curr_out <- tibble(a = curr_a,
                     b = curr_b,
                     product = curr_product,
                     quotient = curr_quotient,
                     alogb = curr_alogb)

  if(i == 1){
    math_out <- curr_out
  }else{
    math_out <- rbind(math_out,
                      curr_out)
  }

}
```

Function alternative:

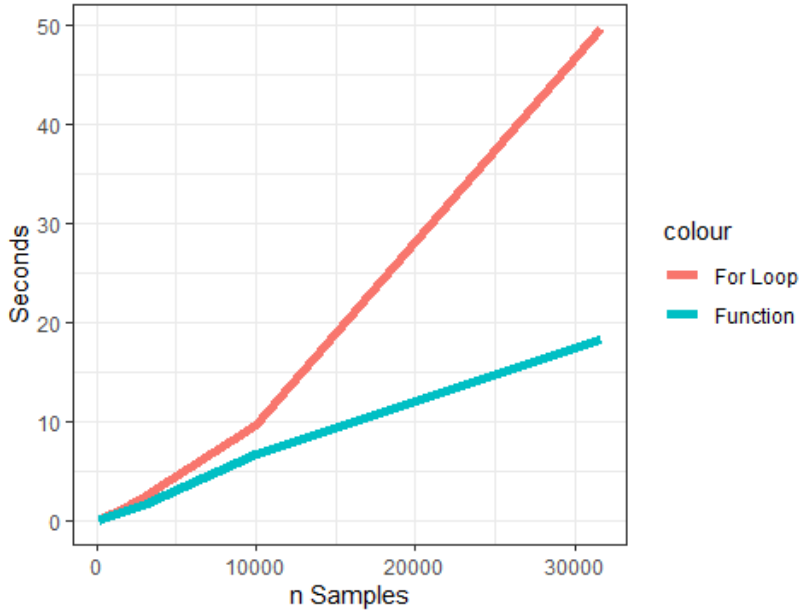
```
math <- function(ab_tib){
  curr_a <- ab_tib[1]
  curr_b <- ab_tib[2]

  curr_product <- curr_a*curr_b
  curr_quotient <- curr_a/curr_b
  curr_exponent <- curr_a^curr_b
  curr_alogb <- log(curr_a, base = curr_b)

  curr_out <- tibble(a = curr_a,
                     b = curr_b,
                     product = curr_product,
                     quotient = curr_quotient,
                     alogb = curr_alogb)
}

math_out2 <- apply(X = ab_tib,
                   MARGIN = 1,
                   FUN = math)


curr_perf<- tibble(n_sample = n_sample,
                   for_time = end-start,
                   fun_time = end2-start2)
```



Publication quality figures without tearing out your hair (using ggplot)

Create a consistent theme

```r
```{r}
fig_dir <- 'figures_lowresolution/'
figure_device <- 'png'
figure_suffix <- '.png'

if(!dir.exists(fig_dir)){
  dir.create(fig_dir)
}

mytheme <- function(){

    theme_bw() %+replace%

    theme(
      panel.grid.major = element_blank(),
      plot.title = element_text(size = 10,
                                hjust = 0,
                                vjust = 2),
      plot.subtitle = element_text(size = 8),
      axis.title = element_text(size = 8),
      axis.text = element_text(size = 8),
      legend.margin = margin(0, 0, 0, 0),
      legend.spacing.x = unit(2, "mm"),
      legend.spacing.y = unit(2, "mm"))
}

title_size <- 10

dpi_figures <- 300

```
```

Use ggsave to output at desired size/dpi

```r
ggsave(filename = paste0(fig_dir, 'figure_1b', figure_suffix),
       plot = f1b,
       width = 3.5,
       height = 3.5,
       device=figure_device,
       dpi=dpi_figures)
```