Exploring CD8+ T-cell Specificities using Single Cell Immune Profiling with an Outlook to Reproducible Bio Data Science

William Hagedorn-Rasmussen

Table of contents

1 Imports

```
library(TCRSequenceFunctions)
library(knitr)
```

2 Introduction

This package, TCRSequenceFunctions, is a collection of functions made for working with data sets from a Single Cell Immune Profiling experiment made by 10x Genomics [1]. There is a total of four data sets which all follow the same general structure. They differ in that, they contain data from each their own respective donor.

The data sets contains binding counts between the donors' library of T-Cell Receptors (TCRs) and a set of peptide-major histocompatibility complexes (pMHCs). The before-mentioned binding counts are so called unique molecular identifier (UMI) counts. For an explanation of all columns see Section 6.

In the following sections, each of the functions contained in TCRSequenceFunctions will be explained, demonstrated and reasoned for. Generally, they can be divided into three types: Cleaning, Augmenting and Modelling where the main goal of the two first is to make the data tidy.

Lastly will be a short section on a Shiny Package, TCRSequenceShiny, which utilizes these functions to make a user-friendly interactive interface for data exploration.

3 Tidying the data

The aim of tidying the data is to enable the data handling, and to ensure a reproducible result. Firstly, the data is cleaned e.g. by making sure, all cells only contain one piece of information. Afterwards, some augmented was needed to enable the modelling. This was done by e.g. adding new columns. A wrapper function was used to run all the preparation functions: `run_all_prep()`. This wrapper simply takes one of the raw data files included in the package as input, and pipe it through all the preparation functions, and output tidy data as in Table 1.

```
data <- data_donor_one_raw_mock %>%
  run_all_prep() %>%
  as.data.frame()

kable(head(data[1:4]))
```

Table 1: Tidy data created from raw file

barcode	TCR_sequence	TCR_combination	donor
TTGCCGTTCCGAATGT-14	CADPSGSARQLTF	one_alpha_one_beta	donor1
TTGCCGTTCCGAATGT-14	CADPSGSARQLTF	one_alpha_one_beta	donor1
TTGCCGTTCCGAATGT-14	CASSQEAGAATGELFF	one_alpha_one_beta	donor1

barcode	TCR_sequence	TCR_combination	donor
TTGCCGTTCCGAATGT-14	CASSQEAGAATGELFF	one_alpha_one_beta	donor1
TTGCCGTTCCGCGCAA-8	CVGDGGSQGNLIF	one_alpha_one_beta	donor1
TTGCCGTTCCGCGCAA-8	CASSEGGFHPLHF	one_alpha_one_beta	donor1

3.1 Cleaning

As mentioned above, cleaning the data is mostly focusing on handling already present data and/or re-structure the data frame. The list of cleaning functions are as follows:

1. `remove_unnecessary_columns()`
2. `split_TCR_sequences_find_non_promiscuous()`
3. `pivot_longer_TCR_sequences()`
4. `add_chain_ident_remove_prefix()`
5. `pivot_longer_pMHC()`
6. `tidy_pMHC_names()`

The first function takes the raw data frame as input, and simply removes the unnecessary columns as these aren't needed. By default, the columns removed are those containing `"_binder"` and the column `"cell_clono_cdr3_nt"`.

`split_TCR_sequences_find_non_promiscuous()` takes the output from the first cleaning function. The purpose is to spil the TCR-sequencs, as to not have cells with multiple pieces of information. Table 2 shows two examples as to how these are written.

```
data <- data_donor_one_raw %>%
  dplyr::select(barcode, cell_clono_cdr3_aa) %>%
  dplyr::sample_n(3) %>%
  as.data.frame()

kable(data)
```

Table 2: A snippet of `data_donor_one_raw` to show two examples as to how TCR-sequences are written

barcode	cell_clono_cdr3_aa
AAAGATGCACCCAGAA-1	TRA:CAVRDRDGGYNKLIF;TRB:CASSQDPSDRPLF
29	
GTCAAGTTCCAGATAA-1	TRA:CAMGTYMNTGFQKLVF;TRA:CAPDRGSTLGRLYF;TRB:CASSRGELGGTDTQ
19	

Table 4: Dimensions of data sets for **(a)** Donor 1, **(b)** Donor 2, **(c)** Donor 3 and **(d)** Donor 4 before and after being prepared by the wrapper `run_all_prep()`

(a)

Donor 1	`raw`	`tidy`
Number of columns	118	27
Number of rows	46526	512264

(b)

Donor 2	`raw`	`tidy`
Number of columns	118	27
Number of rows	77854	860757

(c)

Donor 3	`raw`	`tidy`
Number of columns	118	27
Number of rows	37824	581484

(d)

Donor 4	`raw`	`tidy`
Number of columns	118	27
Number of rows	27308	190482

barcode	cell_clono_cdr3_aa
CAAGATCCAGTTCATA-1	TRA:CATDAEDDKIIF;TRB:CASSLGGWDQPQHF

Table 3: A snippet of `data_donor_one_raw` to show two examples as to how TCR-sequences are written

barcode	**donor**	**cell_clono_cdr3_aa**
AAACCTGAGACAAACA-1	donor1	TRA:CAASVSIWTGTASKLTF;TRA:CAAWDMEYGNKLVF;TRB:CAISDF
AAACCTGAGAGCCCAA-1	donor1	TRA:CASYTDKLIF;TRB:CASSGGSISTDTQYF

3.2 Augmenting

1. `add_max_non_specific_binder()`
2. `evaluate_binder()`
3. `add_TCR_combination_identifier()`

3.3 Change of dimensions

4 Modelling

5 Shiny Integration

References

6 Appendix

Appendix A

[1] 10X Genomics. 2022. A New Way of Exploring Immunity - Linking Highly Multiplexed Antigen Recognition to Immune Repertoire and Phenotype. *10x Genomics* (2022), 1–13. Retrieved from https://www.10xgenomics.com/resources/document-library/a14cde