sciReptor Documentation

Release 1.1.1

Katharina Imkeller, Christian Busse, Francisco Arcila

Mar 31, 2021

CONTENTS:

CHAPTER

ONE

INTRODUCTION

The goal of sciReptor is to be a comprehensive solution for the processing, use and storage of single-cell AIRR-seq (scAIRR-seq) data and its integration with flow cytometry (FC) and receptor affinity (RA) data.

sciReptor was originally developed [Imkeller_2016] to process and analyse data from Matrix PCR [Busse_2014] experiments. Note that sciReptor focuses **exclusively** on single-cell data, if you need to process bulk AIRR-seq data, you will find recommendations on the AIRR-C Software WG website.

sciReptor aims to comply to the AIRR Community standards for metadata [Rubelt_2017], exchange formats [VanderHeiden_2018] and programatic data access [Christley_2020].

It consists out of multiple components:

- Database backend and schema
- Data processing pipeline
- API to access the database via HTTP
- Tools for data import, export and analysis

CHAPTER

TWO

SCIREPTOR DATABASE

sciReptor uses a relational database as backend to store and access data.

2.1 sciReptor schema

2.2 Practical database organization

It is highly recommended to use a separate instance of MariaDB that is dedicated to sciReptor usage. In general, it is assumed that users follow this hierarchy:

- *Database*: One per study, which can encompass multiple *Experiments*
- *Experiment*: A single matrix or a slice of it
- *Run*: The raw data of an individual sequencing library

A newly created *Database* can be named at the users discretion, however the following prefixes should not be used:

- `library_`: This is used internally for the reference library databases
- `scireptor_`: Used for sciReptor internal information and study-level metadata

For performance reasons it is further advisable, that larger studies that contain dozend of *Experiments* should be split into multiple *Databases*.

CHAPTER

THREE

SCIREPTOR PIPELINE

The sciReptor data processing pipeline is the central tool to add previously unprocessed data into the backend database. It performs tag identification, consensus building, VDJ annotation and integration of FC data. Furthermore it produces quality control (QC) reports that can be used to spot potential problems with the data early on.

3.1 Prerequisites

- Software (required)
  - Linux system (tested: CentOS 7)
  - MariaDB database
  - BLAST
  - IgBLAST
  - Muscle
  - RazerS
  - Perl with BioPerl
  - R with BioConductor (flowcore package)
- Software (recommended for pre-processing)
  - PandaSeq
- Configuration
  - MariaDB user account with complete database access
  - *.my.cnf* file holding the credentials

3.2 Basic Setup for a project

It is recommended to keep all data for a specific project within one directory and one corresponding database scheme. Each project directory contains its individual version of the sciReptor code, this was done to make projects independend from each other in terms of code updates. To implement an easy way for updates it is recommended to clone the current version of the sciReptor git repository instead of just copying the script files. It is shown below how to do this.

The *config* file contains all information that is required for data processing. It does **not** contain any metadata, although some fields (e.g. *species*) will overlap with the metadata files.

- Database: The pipeline requires a *Database*, i.e., an instance of the sciReptor database schema [link database chapter] provided by a MariaDB 5.5 instance. The user running the processing needs full access to this *Database*.
- Library: The reference library to be used by Blast and RazorS

3.3 Data pre-processing

sciReptor pipeline expects to ingest reads that cover the complete amplicon. However, Illumina as the most frequently used NGS platform cannot deliver reads that would provide full-length coverage of Ig/TCR V regions. Illumina MiSeq 2x300 bp covers this physically, but the pair-reads need to be assembled first, before they can be used by the pipeline. These pre-processing step are described below.

3.3.1 Requirements

The preprocessing described below assumes that the following tools are install on your computer and are in the `PATH`.

- pandaseq
- bbmap
- ...

It further assumes that the sequence data has undergone basic tests for integrity and experimental quality controls (e.g., with `fastqc`). The median quality for the first read of a read pair should be above Q20 for the entire length, for the second read it should not drop below Q20 before 250 bp.

3.3.2 Paired-read assembly

Table 1: PandaSeq settings (`NULL` indicates that this option is **not** set in the commandline)

species	locus	length_max	length_min	overlap_min	threshold_overlap	reference
mouse	A	560	250	50	0.8	[Ludwig_2019]
mouse	B	560	250	50	0.8	[Ludwig_2019]
mouse	H	NULL	300	50	0.8	[Busse_2014]
mouse	K	NULL	300	50	0.8	[Busse_2014]
mouse	L	NULL	300	50	0.8	[Busse_2014]
human	A	550	300	50	0.8	[Wahl_2021]
human	B	550	300	50	0.8	[Wahl_2021]
human	H	550	320	NULL	NULL	[Murugan_2015]
human	K	550	320	NULL	NULL	[Murugan_2015]
human	L	550	320	NULL	NULL	[Murugan_2015]

3.4 Data processing

- Read numbers should be normalized to cell numbers and tag identification efficiency.
- Performing test processing runs with approx. 500 reads/cell is a quick way to gauge the efficiency of the experiment.

CHAPTER

FOUR

SCIREPTOR TOOLS

This is a collection of tools to import and export data from the database backend from/into an AIRR-C CEF compatible format.

CHAPTER

FIVE

SCIREPTOR API

sciReptor offers an ADC API compliant way to search and access data in its database backend.

CHAPTER

SIX

REFERENCES

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

[Busse_2014] Busse CE *et al*. Single-cell based high-throughput sequencing of full-length immunoglobulin heavy and light chain genes. Eur J Immunol 44:597 (2014). DOI: 10.1002/eji.201343917

[Christley_2020] Christley S *et al*. The ADC API: a web API for the programmatic query of the AIRR Data Commons. Front in Big Data (2020). DOI: 10.3389/fdata.2020.00022

[Imkeller_2016] Imkeller K *et al*. sciReptor: analysis of single-cell level immunoglobulin repertoires. BMC Bioinformatics 17:67 (2016). DOI: 10.1186/s12859-016-0920-1

[Ludwig_2019] Ludwig J *et al*. High-throughput single-cell sequencing of paired TCR and TCR genes for the direct expression- cloning and functional analysis of murine T-cell receptors. Eur J Immunol 49:1269 (2019). DOI: 10.1002/eji.201848030

[Murugan_2015] Murugan R *et al*. Direct high-throughput amplification and sequencing of immunoglobulin genes from single human B cells. Eur J Immunol 45:2698 (2015). DOI: 10.1002/eji.201545526

[Rubelt_2017] Rubelt F *et al*. Adaptive Immune Receptor Repertoire Community recommendations for sharing immune-repertoire sequencing data. Nat Immunol 18:1274 (2017). DOI: 10.1038/ni.3873

[VanderHeiden_2018] Vander Heiden JA *et al*. AIRR Community Standardized Representations for annotated immune repertoires. Front Immunol 9:2206 (2018). DOI: 10.3389/fimmu.2018.02206

[Wahl_2021] Wahl I *et al*. *submitted*