**setup\_results\_directory.sh**

**Purpose**

Creates the directory structure required for organizing TCR analysis outputs for both alpha and beta chains.

Ensures that downstream R scripts and Python/Colab notebooks have pre-defined folders for saving results.

**Directory Structure Created**

**Alpha chain:**

outputs/alpha/tcrdist/

├── alignments/img Plots of sequence alignments

├── cluster\_overlaps Heatmaps and Jaccard index outputs

├── cluster\_sizes Cluster summary statistics

└── diversity Diversity metrics (Shannon, Simpson, etc.)

**Beta chain:**

outputs/beta/tcrdist/

├── alignments/img

├── cluster\_overlaps

├── cluster\_sizes

└── diversity

**Usage**

1. Make script executable:

```bash

chmod +x setup\_tcrdist\_dirs.sh

```

1. Run the script:

```bash

./setup\_tcrdist\_dirs.sh

```

1. Verify that the directories exist:

```bash

tree outputs/

```

**Notes**

* mkdir -p ensures that the script doesn’t fail if directories already exist.
* This script should be run before running R or Python TCR analysis scripts, so all outputs have a consistent location.

**TCR analysis.R**

**Purpose**

This script performs TCR repertoire analysis across multiple mouse samples. It:

* + - * Reads in annotated TCR sequencing data.
      * Filters for a specific chain (e.g., beta).
      * Normalizes read counts using TMM normalization (edgeR).
      * Computes intra-mouse and inter-mouse repertoire convergence at nucleotide and amino acid levels.
      * Summarizes overlap patterns and generates plots.
      * Writes out detailed tables of overlapping TCRs per mouse group.

**Inputs**

1. Parameters file:
   * + - repertoires.txt (tab-delimited) with at least these columns:
         * path → file path to each repertoire table
         * mouse → mouse identifier (e.g., F0, N2, BALB\\_C, CBA)
         * tumour → tumour ID or condition label
         * chain → chain type (alpha, beta, etc.)
2. Repertoire files:
   * + - One per row of repertoires.txt.
       - Each should be tab-delimited and contain (at minimum):
         * v\_call, j\_call
         * junction (nucleotide)
         * junction\_aa (amino acid)
         * duplicate\_count (read counts).
3. Configuration inside the script:
   * + - chain\_type = 'beta' → which TCR chain to analyze.
       - write\_files = T → whether to output CSVs and plots.
       - Working directory is set by setup\_session().

**Outputs**

All outputs are written to an outputs/<chain\_type>/ directory (e.g., outputs/beta/).

1. Processed data tables (CSV):
   * + - 0\_complete-data.csv → all processed TCRs with overlap annotations.
       - Subsets filtered by which mouse groups a TCR is present in, e.g.:
         * 1.0\_F0\_only.csv
         * 1.1\_N2\_only.csv
         * 2.0\_F0\_N2.csv
         * 3.2\_BALBC\_CBA\_N2.csv
         * 4.0\_BALBC\_CBA\_F0\_N2.csv
       - Naming convention: <overlap-level>\_<mouse-groups>.csv.
2. Plots (PDF):
   * + - inter-mouse-overlap.pdf → barplot showing the number of overlapping TCRs across mice, colored by overlap size.

**Dependencies**

The script loads the following R packages:

* + - * data.table
      * tidyverse
      * patchwork
      * msa
      * igraph
      * edgeR
      * rlang

**TCRdist.ipynb**

**Purpose**

Generates TCR-specific pairwise distance matrices and probabilities of generation (pgen) for TCR alpha or beta chains using tcrdist3.

This is useful for downstream analyses such as clustering, repertoire overlap, and diversity metrics.

**Key Features**

* + - * Computes TCRdist matrices for each input CSV file.
      * Computes pgen values for each CDR3 sequence using the Olga model.
      * Supports both alpha and beta chains (run in separate code blocks).
      * Outputs compressed results (tar.gz) ready for download.

**Inputs**

* + - * CSV files containing TCR repertoire data from your outputs/{CHAIN} folder.
* Must include at minimum:
  + v\_call → V gene call
  + j\_call → J gene call
  + junction\_aa → CDR3 amino acid sequence
* No need for a count column; script defaults missing counts to 1.

**Outputs**

Saved in a tcrdist folder within Colab:

1. pgen files:

<original\_filename>\_pgens.csv

Includes computed generation probabilities for each CDR3 sequence.

1. Distance matrices:

<original\_filename>\_tcrdistmatrix.csv.gz

Pairwise TCRdist between sequences in the CSV.

1. Compressed archive for download:

tcrdist.tar.gz

**Instructions for Use**

1. Upload CSV files from outputs/{CHAIN}/\*.csv into Colab.
2. Create output folder:

```bash

!mkdir tcrdist

```

1. Install tcrdist3 and load dependencies:

```python

!pip install tcrdist3

import pandas as pd

from tcrdist.pgen import OlgaModel

from tcrdist.repertoire import TCRrep

import os, re

from google.colab import files

```

1. Run beta chain block or alpha chain block.

* Beta: maps columns to v\_b\_gene, j\_b\_gene, cdr3\_b\_aa
* Alpha: maps columns to v\_a\_gene, j\_a\_gene, cdr3\_a\_aa

1. After processing, download results archive:

```python

files.download("tcrdist.tar.gz")

```

6. To process the other chain, remove old files except for any pre-loaded sample data and repeat step 4.

**Notes**

* Fake allele \*01 is appended to gene names for TCRdist compatibility.
* Warnings about missing counts are normal; each clone is assigned a count of 1 by default.
* Distance matrices are gzip-compressed CSVs to reduce file size.
* Compatible with mouse repertoires (organism set to "mouse").

**TCR clustering.R**

**Purpose**

This script clusters TCR sequences based on TCRdist matrices and performs detailed cluster analysis. Specifically, it:

* + - * Loads pre-computed TCRdist distance matrices and associated annotated repertoire data.
      * Performs hierarchical clustering at a user-defined threshold (default h = 55).
      * Summarizes cluster size distributions and fits them to a gamma distribution.
      * Produces per-cluster statistics (gene usage, number of unique CDR3s, generation probability distributions).
      * Generates sequence logos for representative clusters.
      * Outputs processed data tables, histograms, summary statistics, and cluster alignments.

**Inputs**

1. Working directory: set in

setup\_session('~/OneDrive - University College London/\_Leo Post Doc/\_Ari-TCR-analysis/')

1. Base directory (per chain):
   * Defined as outputs/<chain>/tcrdist/
   * Example: outputs/alpha/tcrdist/
2. Input files inside this directory:
   * + - <sample>\_pgens.csv → annotated repertoire data with columns such as mouse, v\_\*\_gene, j\_\*\_gene, cdr3\_\*\_aa, pgen, etc.
       - <sample>\_tcrdistmatrix.csv → pairwise TCRdist distance matrix for the same repertoire.

**Outputs**

All results are written under outputs/<chain>/tcrdist/, with subfolders for organization.

1. Clustered data (per repertoire):

* <sample>\_pgens\_clustered.csv → original repertoire annotated with cluster\_id.

1. Cluster size histograms:

* cluster\_sizes/<sample>-cluster-size-histogram.pdf
* Histogram of cluster sizes, overlaid with fitted gamma distribution curve (if fitting successful).

1. Cluster summary statistics (per repertoire):

* cluster\_sizes/<sample>-cluster-summary-stats.csv
* Contains, for each cluster:
  + cluster\_id
  + cluster\_count (number of members)
  + mice\_shared (unique mice represented)
  + v\_genes / j\_genes used
  + unique\_cdr3s (amino acid diversity)
  + mean\_pgen and pgen\_skew (generation probability statistics).

1. Cluster alignments (selected clusters):
   * FASTA files: alignments/<sample>-cluster<N>.fasta (aligned CDR3 sequences).
   * Sequence logos (PDFs): alignments/img/<sample>-cluster<N>.pdf.
   * By default, either:
     + all clusters with ≥10 members, or
     + top 10 largest clusters (if none ≥10).

**Parameters**

* chain → which TCR chain to analyze (alpha or beta).
* cut\_at → clustering height threshold (default 55).
* write\_aln → whether to write alignments and logos (TRUE/FALSE).

**Dependencies**

The script loads the following R packages:

* data.table
* tidyverse
* patchwork
* msa (for sequence alignment)
* igraph
* edgeR
* rlang
* fitdistrplus (distribution fitting)
* e1071 (skewness)
* ggseqlogo (sequence logos)

**TCR diversity.R**

**Purpose**

Generates plots and diversity metrics from processed TCR repertoires (clustered or clonotype-level). Specifically, it:

* Visualizes identical TCR sharing within and between mice (nucleotide and amino acid sequences).
* Computes diversity metrics per mouse and replicate using:
  + Shannon diversity (raw and normalized by max possible)
  + Simpson diversity
  + Chao1 richness estimate
* Applies metrics to both TCRdist clusters and amino acid clonotypes.
* Outputs plots as PDFs and diversity tables as CSVs for downstream analyses.

**Inputs**

1. Processed clustered TCR data:

outputs/<chain>/tcrdist/0\_complete-data\_pgens\_clustered.csv

* Must include columns:
  + mouse (F0, N2, BALBC, CBA)
  + B1, B2, B3 (replicate read counts)
  + id\_aa (amino acid clonotype ID)
  + cluster\_id (TCRdist cluster ID).

1. Parameters in script:
   1. chain\_type → which TCR chain to analyze (alpha or beta).
   2. write\_loc → directory for saving outputs (outputs/<chain>/tcrdist/diversity/).

**Outputs**

1. Diversity tables (CSV):
   1. Saved to outputs/<chain>/tcrdist/diversity/
   2. Filenames include \*\_diversity.csv corresponding to:
      * tcrdist\_reads → Shannon/Simpson for TCRdist cluster read proportions
      * tcrdist\_size → Shannon/Simpson/Chao1 for TCRdist cluster sizes
      * clonotype\_reads → Shannon/Simpson for amino acid clonotype read proportions
      * clonotype\_size → Shannon/Simpson/Chao1 for amino acid clonotype sizes
2. Plots (PDF):

* diversity\_plots.pdf (multi-page)
* Includes:
  + Barplots of identical TCR sharing (nucleotide and amino acid)
  + Boxplots of diversity metrics (Shannon, normalized Shannon, Simpson)
  + Separate visualizations for clusters (tcrdist) and amino acid clonotypes

**Dependencies**

Requires R packages:

* data.table
* tidyverse
* patchwork
* msa
* igraph
* edgeR
* rlang
* vegan

**TCR overlap.R**

**Purpose**

Analyzes TCR sharing between mice at both TCRdist cluster and amino acid clonotype levels. Specifically, it:

* Generates presence-absence heatmaps of clusters/clonotypes across mice.
* Calculates pairwise Jaccard indices for mouse repertoires.
* Produces combined visualizations (heatmaps + Jaccard indices).
* Exports both data tables and plots for downstream analyses.

**Inputs**

* + - 1. Cluster summary statistics:

outputs/<chain>/tcrdist/cluster\_sizes/0\_complete-data\_cluster-summary-stats.csv

Must include:

* cluster\_id
* mice\_shared → comma-separated list of mice sharing the cluster
  + - 1. Clustered repertoire data:

outputs/<chain>/tcrdist/0\_complete-data\_pgens\_clustered.csv

**Must include:**

* mouse
* aa\_present\_in → comma-separated list of mice sharing the amino acid clonotype
  + - 1. Parameters in script:
* chain\_type → "alpha" or "beta"

**Outputs**

Saved under: outputs/<chain>/tcrdist/cluster\_overlaps/

1. Presence-absence tables (CSV):

* <source>\_TCRdist\_presence-absence\_data.csv
* <source>\_AA\_clonotype\_presence-absence\_data.csv

1. Jaccard index tables (CSV):

* <source>\_TCRdist\_jaccard-index\_data.csv
* <source>\_AA\_clonotype\_jaccard-index\_data.csv

1. Plots (PDF):

* Jaccard indices → <source>\_TCRdist\_jaccard-index\_plots.pdf, <source>\_AA\_clonotype\_jaccard-index\_plots.pdf
* Presence-absence heatmaps → <source>\_presence-absence\_heatmaps.pdf

**Dependencies**

Requires R packages:

* tidyverse
* patchwork