TraCeR: TCR α and β Chain Sequencing to Determine Clonotype from C1 Single-Cell mRNA Seq Whole Transcriptome Data

Introduction

TraCeR is a computational method to reconstruct full-length, paired T cell receptor (TCR) sequences from T lymphocyte single-cell RNA sequence data. This powerful tool allows linkage of T cell specificity with functional response by revealing clonal relationships between cells along with each cell's transcriptional profile. The *Nature Methods* paper describing TraCeR in detail can be found here.

The application described here uses the Fluidigm® C1™ Single-Cell mRNA Seq Protocol (PN 100-7168) to provide whole transcriptome amplification (WTA) with full-length mRNA transcripts. Conventional whole transcriptome data analysis approaches do not account for highly variable regions such as recombined TCR sequences because such sequences are not present within reference transcriptomes. TraCeR analysis allows reconstruction of the variable sequence of TCR α and β chains through use of a "combinatorial recombinome" library of all possible TCR sequences. Libraries for both mouse and human V(D)J are available through TraCeR.

TraCeR may be applied in parallel alongside whole transcriptome mRNA seq analysis to provide phenotypic and clonotypic profiling of T cell populations, linking transcriptional status to receptor sequence.

Resources for implementing the single-cell TCR sequencing protocol are freely available at Script Hub™: fluidigm.com/c1openapp/scripthub/featured/tcr-sequence-determination. Figure 1 describes the workflow:

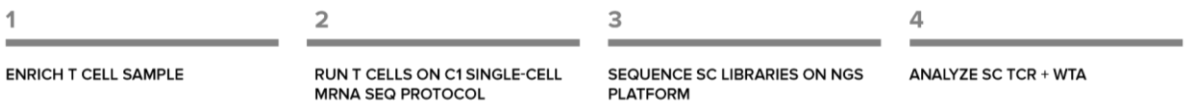

Figure 1. Overall workflow for using C1 to obtain single-cell (SC) TCR and WTA from T cell mRNA seq libraries. The T cell-enriched samples are run on the C1 with the mRNA Seq protocol (PN 100-7168). After next-generation sequencing (NGS) of the single-cell libraries, the TraCeR analysis identifies α and β chain sequences at the single-cell level. Further analysis enables correlation of clonotype with the single-cell phenotypic characterization obtained from WTA.

Background

The immune system employs T cells to survey other cells in the body for the presentation of antigenic peptides. In these cell-cell interactions the TCR binds to a peptide-major histocompatibility complex (pMHC) presented on the surface of an antigen-presenting cell (APC), resulting in T cell activation and proliferation. The TCR antigenic binding site, shown in Figure 2, is encoded by highly variable DNA sequences, leading to a high level of receptor diversity across the T cell population. This provides breadth and specificity to T cell activation by antigen recognition.

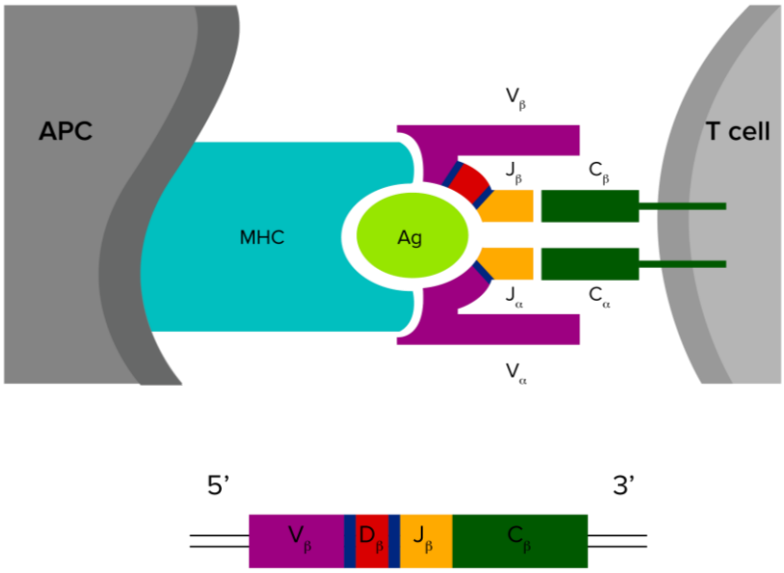

Figure 2. The TCR binds to peptides from antigen-presenting cells (APC) to identify immunological threats. The hypervariable regions (V, D, and J) in the TCR α and β chains create diversity in TCR antigen specificity, creating the T cell repertoire. Hypervariable regions in the α and β chain mRNA transcripts are created during T cell development by recombination of TCR genes from germline DNA. The constant region is denoted as C.

The pMHC-binding region of the TCR is determined by the pairing of two protein chains, α and β; each chain undergoes genetic recombination during T cell development to generate this variable sequence (Figure 2). Each recombined sequence at the single-cell level will be comprised of randomly chosen gene segments (VD for α chain or VDJ for β chain). VDJ refers to the following sequence of gene segments: V (variable), D (diversity), J (joining). The resulting sequences of both the α and β chains at the single-cell level determines the pMHC specificity of the receptor.

Due to the high number of possible recombinant sequences, it is extremely unlikely that two T cells with identical TCR DNA sequences will develop in the thymus. Once in the periphery, however, interaction of a T cell clone with its cognate pMHC may cause the clone to proliferate. This results in an expansion of that particular T cell clonotype,

with all progeny sharing the same TCR and thus specificity to the same antigen. This expanded clonotype includes phenotypically different subsets of T cells that are part of the orchestrated immune response.

The ability to map the TCR sequences of individual T cells, identify clonotype expansion, and link that to phenotype provides a high level of resolution when studying the immune response to cancer, infection, and autoimmune disorders—resolution that cannot be derived from bulk-cell analysis. This single-cell resolution also enhances the basic understanding of immune repertoire evolution and function and is critical in therapeutic TCR molecule engineering.

The Advantages of Single-Cell, Paired Alpha-Beta Chain TCR Analysis and Whole Transcriptome RNA Seq

The application described here (Stubbington *et al*.) enables T cell clonotypes to be linked to cell phenotype and function through whole transcriptome sequencing, including α and β chain pairing, at the single-cell level. The use of first-strand synthesis, with oligo dT and template switching for full-length transcript, allows comprehensive sequencing of the highly variable region of the TCR. By sequencing with single-base resolution, one is able to elucidate the random insertion and deletion of nucleotides that create the junctional diversity of the V(D)J region.

While targeted sequencing approaches can also be used to derive the TCR α and β chain sequences, these require careful and time-consuming protocols and only provide limited accompanying phenotypic information. High-throughput, single-cell sequencing approaches that employ barcoding strategies and 3′ or 5’ end counting to streamline the workflow do not provide the detailed sequence information across the V(D)J region that is essential to identify the clonotype of single cells.

Key advantages of the method outlined here are:

- Simple workflow using validated C1 mRNA Seq protocol and reagents
- Single-cell, paired α and β chain TCR sequence information
- Unbiased, comprehensive gene expression information for each cell
- Detailed outline of the analysis pipeline, TraCeR, to link single-cell clonotype to phenotypic profile

Sequencing Depth Guidance

The data presented in Stubbington et al. was sequenced with paired-end 100-base reads. The sequencing depth analysis indicated that 1 million reads per cell is sufficient to detect 92% of TCR α and β chain sequences in single mouse lymphocytes.

TraCeR Workflow for TCR Reconstruction from C1 mRNA Seq Datasets

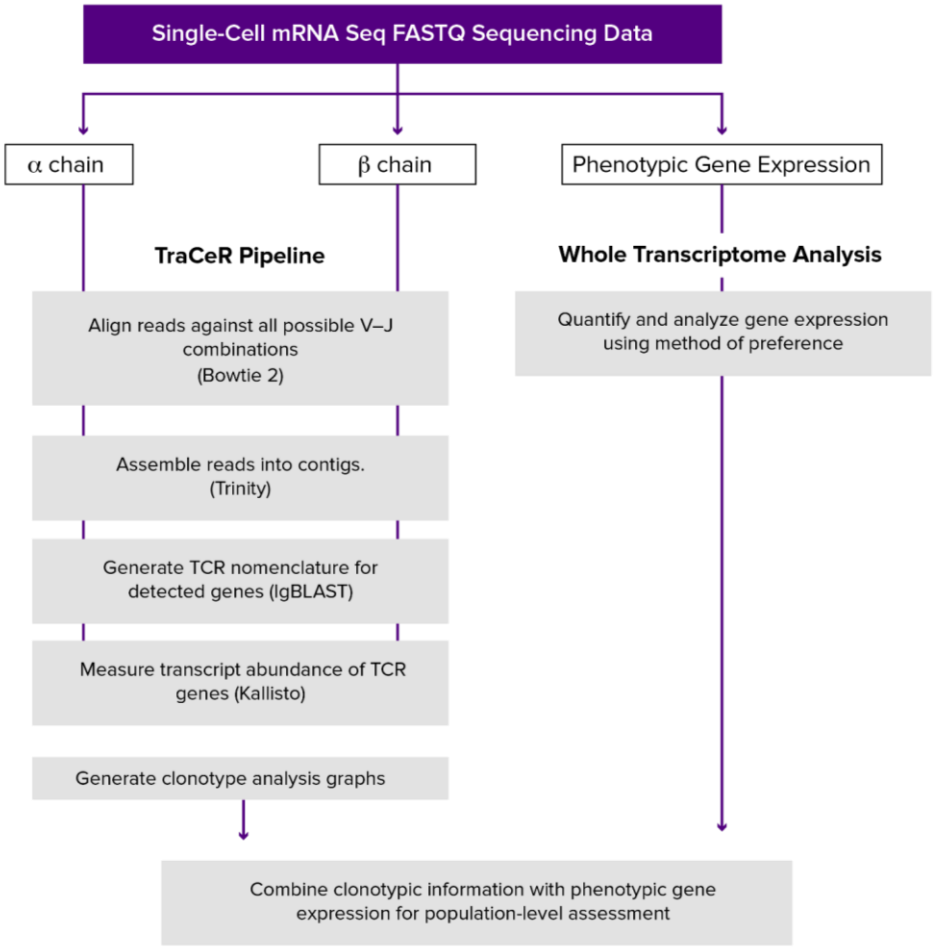

Figure 3. TraCeR analysis workflow for combined analysis of TCR and phenotypic gene expression from whole transcriptome data. The analysis has three parallel tracks. The data is analyzed for the presence of α and β chains with the TraCeR pipeline. Each chain has a reference set of "combinatorial recombinomes" for the respective analysis. The third track is the analysis of phenotypic gene expression. The three components, α and β chains, and the aggregate phenotypic gene expression are then combined to correlate the TCR clonotype with T cell phenotype.

The TraCeR analysis pipeline uses several pre-existing tool sets to extract, reconstruct, and assign TCR sequence information and nomenclature to single-cell mRNA seq data.

Bowtie 2

The TraCeR TCR reconstruction tool extracts TCR-derived sequencing reads for each cell by alignment against "combinatorial recombinomes" comprising all possible combinations of V and J segments. This dataset is used to extract the respective α and β chain reads from the sample. The nucleotide sequence at the joining of the VJ or DJ genes is comprised of a short random sequence that is not encoded within genomic DNA. To enable the alignment over these joining regions (referred to as ambiguous alignment regions because of their unspecified sequence), "N" nucleotides are placed in the synthetic genome VD and VDJ sequences.

Bowtie 2, a tool for aligning sequencing reads to long reference sequences, is used because it can align against ambiguous nucleotides. It will perform gapped alignments with gaps in both the read and the reference sequence. Low penalties are applied for such alignments to ensure maximum sensitivity.

The alignment criteria are specified in Table 1. A review of the input criteria for Bowtie 2 is available at:

Bowtie-bio.sourceforge.net/bowtie2/manual.shtml#bowtie2-options-np

Table 1. Bowtie 2 Alignment Criteria

Sequence Alignment/Map Format (SAM) and Scoring Options	Values
--no-unal	–k 1
--np	0
--rdg	1,1
--rfg	1,1

Trinity and IgBLAST

The aligned reads are assembled into contigs using the RNA seq *de novo* assembler Trinity. IgBLAST is then used to analyze each TCR contig and determine the V, D, and J segments along with the nucleotide sequences of the junctions. The contigs are screened for correct locus assignment and for an E-value $<5 \times 10^{-3}$. Where more than one contig in a cell is derived from the same recombinant, the results are combined and the longest contig (i.e., the one with most information) is used to determine the segment usage. After locus assignment, the recombinant sequence is given an identifier that contains the V variable gene name, junctional nucleotide sequence, and J joining gene name (e.g., TRBV31_AGTCTTGACACA AGA_TRBJ25).

Kallisto

The transcriptome expression levels of TCR genes are determined with the pseudo-alignment-based Kallisto algorithm. Detected recombinant sequences are appended to the mouse transcriptome prior to Kallisto index generation and transcript quantification using the RNA seq reads from the cell in question.

TraCeR Graphics

The last step of the TraCeR analysis generates clonotypic bar graphs, reconstructed α and β chain length distributions, and TCR sequence-network graphs via custom Python scripts. The clonotypic bar graphs provide a measure of the clonotype diversity and the number of cells detected for each clonotype. The network graphs show shared TCR sequences across the single-cell population measured with C1.

Phenotypic Gene Expression Analysis and Clonotype Association

In Stubbington et al., additional single-cell gene expression was measured from the same sequencing data using a parallel analysis with GSNAP for alignment (research-pub.gene.com/gmap/) and HTSeq, a Python-based sequence analysis platform. Any appropriate method can be applied to quantify overall gene expression.

Reduced dimensionality analysis was performed with independent component analysis using the Fast ICA Python package. This allowed the positions of clonally related cells to be visualized within the gene expression space.

The method described in this application note generates standard whole transcriptome RNA seq data and so is amenable to any downstream gene expression analysis and clustering techniques.

Setting Up the TraCeR Environment

Detailed instructions for setting up the TraCeR environment are available in Appendix 1. The following software versions have been verified for a successful environment setup in Ubuntu version 14.04.

Software Dependencies Installed with apt-get Package Manager (Current as of March 2016)

build-essential

graphviz

libncurses5-dev

python-biopython

python-Levenshtein

python-matplotlib

python-NetworkX

python-NumPy

python-pandas

python-PrettyTable

python-SciPy

zlib1g-dev

Software Versions

Bowtie 1: bowtie-1.1.2-linux-x86_64

- bowtie-bio.sourceforge.net/index.shtml

Bowtie 2: bowtie2-2.2.7-linux-x86_64

- bowtie-bio.sourceforge.net/index.shtml

igBLAST: ncbi-igblast-1.4.0-x64-linux

Kallisto: kallisto_linux-v0.42.4

Python: 2.7.6

Reference Human DNA: Homo_sapiens.GRCh38.rel79.cdna.all.fa.gz

Reference Mouse DNA: us_musculus.GRCm38.rel79.cdna.all.fa.gz

SAMtools: samtools-1.3.tar.bz2

Seaborn: 0.7.0

Trinity: trinityrnaseq_r20140717

References

"Using C1 to Generate Single-Cell cDNA Libraries for mRNA Sequencing." Fluidigm PN 100-7168

Stubbington, M.J., Lönnberg, T., Proserpio, V. et al. "T cell fate and clonality inference from single-cell transcriptomes." *Nature Methods* 13 (2016): 329–32.

Support

Please email questions or report problems to ms31@sanger.ac.uk

Appendix 1: Detailed Instructions for TraCeR Setup

The following text is taken from the ReadMe doc at this link:
github.com/Teichlab/tracer/blob/master/README.md

Installation

TraCeR is written in Python and can be downloaded, made executable (with `chmod u+x tracer`), and run. Or it can be simply run with `python tracer`.

Download the latest version and accompanying files from www.github.com/teichlab/tracer.

Prerequisites

TraCeR relies on several additional tools and Python modules.

Software

1 Bowtie 2—Required for alignment of reads to synthetic TCR genomes.
bowtie-bio.sourceforge.net/index.shtml

2 Trinity—Required for assembly of reads into TCR contigs.
sourceforge.net/projects/trinityrnaseq/files/PREV_CONTENTS/previous_releases/

IMPORTANT Currently TraCeR uses Trinity parameters intended for use with Trinity v1. Updates for use with Trinity v2 are coming soon.

NOTE Trinity requires a working installation of Bowtie v1.
bowtie-bio.sourceforge.net/index.shtml

3 IgBLAST—Required for analysis of assembled contigs
ftp.ncbi.nih.gov/blast/executables/igblast/release/.

4 Kallisto—Required for quantification of TCR expression.
pachterlab.github.io/kallisto/

5 Graphviz—Dot and neato drawing programs required for visualization of clonotype graphs.
graphviz.org/

Fast ICA is not part of TraCeR but is used for independent component analysis plots of the phenotypic and clonotypic data.
pypi.python.org/pypi/MDP/2.6

Note on Installing IgBLAST

In addition to downloading the IgBLAST executable files from `ftp.ncbi.nih.gov/blast/executables/igblast/release/<version_number>`, the internal data directory must be downloaded (`ftp.ncbi.nih.gov/blast/executables/igblast/release/internal_data`) and placed in the same directory as the IgBLAST executable files. This is required for a working IgBLAST installation and is also described in the IgBLAST README file.

The $IGDATA environment variable must point to the location of the IgBLAST executable. For example, run `export IGDATA=/<path_to_igblast>/igblast/1.4.0/bin.`

Python Modules

- Biopython
- Levenshtein
- Matplotlib
- NetworkX

  NOTE If using NetworkX v1.11 or later, pydotplus must also be installed, for writing dot files for use with Graphviz.
- PrettyTable
- Seaborn

Setup

The python modules can be installed by running the setup script:

```
python setup.py install
```

Or using pip from the requirements.txt file:

```
pip install -r requirements.txt
```

Once the prerequisites are installed, TraCeR must be set up with a corresponding configuration file.

TraCeR uses a configuration file to point it to the locations of files that it accesses. By default, this is `tracer.conf` and is in the same directory as the TraCeR executable. The `-c` option to the various tracer modules allows user specification of another file for the configuration file.

IMPORTANT If relative paths in the config file are specified, these will be used as relative to the directory that contains the `tracer` executable.

External Tool Locations

Editing `tracer.conf` (or a copy) is necessary to set the paths within the `[tool_locations]` section to point to the executables for all of the required tools.

```
[tool_locations]
#paths to tools used by TraCeR for alignment, quantitation, etc
bowtie2_path = /path/to/bowtie2
igblast_path = /path/to/igblastn
kallisto_path = /path/to/kallisto
trinity_path = /path/to/trinity
dot_path = /path/to/dot
neato_path = /path/to/neato
```

Resource Locations and Necessary Files

The tools used by TraCeR need a variety of additional files to work properly and to allow extraction of TCR-derived reads, expression quantification, etc. The locations of these files are specified in the other sections of the configuration file and are detailed below.

Currently, organism-specific files (TCR gene sequences, synthetic genome indices, igblast_indices) for mouse and human are distributed with the source code in the resources directory.

Bowtie Synthetic Genomes Path

```
[bowtie2_options]
synthetic_genome_index_path = resources/synthetic_genomes/mouse
```

This path specifies the directory that contains Bowtie 2 indices constructed from all possible combinations of V and J segments for each locus.

Trinity Options

Jellyfish Memory

```
[trinity_options]
```

To specify the maximum memory for Trinity Jellyfish should be set appropriately for the given environment. An example of the specification is given as follows.

```
max_jellyfish_memory = 1G
```

HPC Configuration

```
trinity_grid_conf = /path/to/trinity/grid.conf
```

Trinity can parallelize contig assembly by submitting jobs across a compute cluster. If running in such an environment, specify an optional Trinity config file here. Additional information is available in the Trinity documentation.

IgBLAST Options

Databases Path

```
[IgBlast_options]
igblast_index_location = resources/igblast_dbs/mouse
```

VDJ Sequences

This path specifies the directory that contains IgBLAST database files for V, D, and J genes. These files are named `imgt_tcr_db_<SEGMENT>.fa`.

```
imgt_seq_location = resources/imgt_sequences/mouse
```

Files are named by the following convention:

`TR<LOCUS><SEGMENT>.fa`.

Receptor Type

```
igblast_seqtype = TCR
```

NOTE TraCeR currently works only with TCR sequences.

Kallisto Options

```
[kallisto_options]
base_transcriptome = /path/to/kallisto/transcriptome
```

Location of the transcriptome fasta file to which the specific TCR sequences will be appended from each cell. The file can be downloaded from bio.math.berkeley.edu/kallisto/transcriptomes/. This must be a plain text fasta file that may need to be decompressed (files from the Kallisto link are gzipped).

Testing TraCeR

TraCeR includes a small (three-cell) dataset for mouse α and β chain sequences in test_data/ . This can be used to test the installation and config file and confirm that all the prerequisites are working.

Run as:

```
tracer test -p <ncores> -c <config_file>
```

This will perform the `assemble` step using the small test dataset. It will then perform `summarise` using the assemblies that are generated along with precalculated output for two other cells (in `test_data/results`).

Compare the output in `test_data/results/filtered_TCR_summary` with the expected results in `test_data/expected_summary`. There should be three cells, each with one productive alpha, one productive beta, one nonproductive alpha, and one nonproductive beta. Cells 1 and 2 should be in a clonotype.

Using TraCeR

Tracer has two modes, `assemble` and `summarise`, that are run in sequence.

`Assemble` reads FASTQ files of paired-end RNA seq reads from a single cell and reconstructs TCR sequences.

`Summarise` accesses a set of directories containing output from the *assemble* phase (each directory represents a single cell) and summarizes TCR recovery rates and generates clonotype networks.

Assemble: TCR Reconstruction

Usage

```
tracer assemble [options] <file_1> [<file_2>] <cell_name>
<output_directory>
```

Main Arguments

- `<file_1>` : FASTQ file containing #1 mates from paired-end sequencing or all reads from single-end sequencing.
- `<file_2>` : FASTQ file containing #2 mates from paired-end sequencing. Not for data from single-end sequencing.
- `<cell_name>` : Name of the cell. This is arbitrary text that will be used for all subsequent references to the cell in filenames/labels, etc.
- `<output_directory>` : Directory for output; will be created if it does not exist. Cell-specific output will go into `/<output_directory>/<cell_name>`. This path should be the same for every cell that is summarized together.

Options

- `-p/--ncores <int>` : Number of processor cores available. This is passed to Bowtie 2 and Trinity. Default=1.
- `-c/--config_file <conf_file>` : Config file to use. Default = `tracer.conf`
  `-s/--species` : 10 summarise step because it defines the V segments that are indicative of iNKT cells. Default = Mmus.
- `-r/--resume_with_existing_files` : If this is set, TraCeR will search for existing output files and not rerun steps that appear to have been completed. This saves time if TraCeR dies partway through a step, allowing the program to resume at the same point.
- `-m/--seq_method` : Method to generate sequences for assessment of recombinant productivity. By default (`-m imgt`), TraCeR replaces all but the junctional sequence of each detected recombinant with the reference sequence from IMGT prior to assessing

productivity of the sequence. This makes the assumption that sequence changes outside the junctional region are due to PCR/sequencing errors rather than being genuine polymorphisms. This is likely to be true for well-characterized mouse sequences but may be less so for human and other outbred populations. To determine productivity from only the assembled contig sequence for each recombinant, use `-m assembly`.

- `--single_end` : Use this option if the data are single-end reads. If this option is set, fragment length and fragment sd must be specified as indicated below.
  `--fragment_length` : Estimated average fragment length in the sequencing library, used for Kallisto quantification. Required for single-end data. Can also be set for paired-end data to avoid the estimate by Kallisto.
- `--fragment_sd` : Estimated standard deviation of average fragment length in the sequencing library used for Kallisto quantification. The value is required for single-end data but can also be set for paired-end data to avoid the estimate by Kallisto.

Output

For each cell, an `/<output_directory>/<cell_name>` directory will be created. This will contain the following subdirectories.

1. `<output_directory>/<cell_name>/aligned_reads`
   Contains the output from Bowtie 2 with the sequences of the reads that aligned to the synthetic genomes.

2. `<output_directory>/<cell_name>/Trinity_output`
   Contains fasta files for each locus where contigs could be assembled and two text files that log successful and unsuccessful assemblies.

3. `<output_directory>/<cell_name>/IgBLAST_output`
   Contains files with the output from IgBLAST for the contigs from each locus.

4. `<output_directory>/<cell_name>/unfiltered_TCR_seqs`
   Contains files describing the TCR sequences that were assembled prior to filtering by expression, if necessary.
   - `unfiltered_TCRs.txt` : Text file containing TCR details. The file begins with the count of productive/total rearrangements detected for each locus, then has details of each detected recombinant.
   - `<cell_name>_TCRseqs.fa` : fasta file containing full-length, followed by reconstructed, TCR sequences.
   - `<cell_name>.pkl` : Python pickle file containing the internal representation of the cell and its recombinants as used by TraCeR. This is used in the summarization steps.

5 `<output_directory>/<cell_name>/expression_quantification`
Contains Kallisto output with expression quantification of the entire transcriptome, including the reconstructed TCRs.

6 `<output_directory>/<cell_name>/filtered_TCR_seqs`
Contains the same files as the unfiltered directory above, but these recombinants have been filtered so that only the two most highly expressed sequences from each locus are retained. This resolves biologically implausible situations where more than two recombinants are detected for a locus. **This directory contains the final output with high-confidence TCR assignments**.

Summarize: Summary and Clonotype Networks

Usage

```
tracer summarise [options] <input_dir>
```

Main Argument

`<input_dir>` : Directory containing subdirectories of each cell to summarize.

Options

- `-c/--config_file <conf_file>` : config file to use. Default = tracer.conf.
- `-u/--use_unfiltered` : This flag should be set to use unfiltered recombinants for summary and networks rather than the recombinants filtered by expression level.
- `-i/--keep_inkt` : TraCeR may identify iNKT cells by their characteristic TCRA gene segments (TRAV11–TRAJ18). By default, these are removed before creation of clonotype networks. Setting this option retains the iNKT cells in all stages.
- `-g/--graph_format` : Output format for the clonotype networks. This is passed directly to Graphviz and so must be one of the options detailed at graphviz.org/doc/info/output.html.

Output

Output is written to `<input_dir>/filtered_TCR_summary` or `<input_dir>/unfiltered_TCR_summary` depending on whether the `--use_unfiltered` option was set.

The following output files are generated:

1 `TCR_summary.txt` Summary statistics describing successful TCR reconstruction rates and the numbers of cells with 0, 1, 2, or more recombinants for each locus.

2 `recombinants.txt` List of TCR identifiers, lengths, and productivities for each cell.

3 `reconstructed_lengths_TCR[A|B].pdf` and `reconstructed_lengths_TCR[A|B].txt` Distribution plots (and text files with underlying data) showing the lengths of the VDJ regions from assembled TCR contigs. Longer contigs give higher-confidence segment assignments. Text files are only generated if at least one TCR is found for a locus. Plots are only generated if at least two TCRs are found for a locus.

4 `clonotype_sizes.pdf and clonotype_sizes.txt` Distribution of clonotype sizes in bar graph and text file formats.

5 `clonotype_network_[with|without]_identifiers.<graph_format>` Graphical representation of clonotype networks either with full recombinant identifiers or simple lines indicating presence or absence of recombinants.

6 `clonotype_network_[with|without]_identifiers.dot` Files describing the clonotype networks in the Graphviz dot language.

CORPORATE HEADQUARTERS
7000 Shoreline Court, Suite 100
South San Francisco, CA 94080 USA
Toll-free: 866 359 4354 in the US
Fax: 650 871 7152
fluidigm.com

SALES
North America | +1 650 266 6170 | info-us@fluidigm.com
Europe/EMEA | +33 1 60 92 42 40 | info-europe@fluidigm.com
China (excluding Hong Kong) | +86 21 3255 8368 | info-china@fluidigm.com
Japan | +81 3 3662 2150 | info-japan@fluidigm.com
All other Asian countries | +1 650 266 6000 | info-asia@fluidigm.com
Latin America | +1 650 266 6000 | info-latinamerica@fluidigm.com