

TranD User Guide

Table of Contents:

TranD Overview 3

***Transcriptome Consolidation* 4**

***Comparison of Two Transcriptomes* 6**

**Clone (copy) TranD from github and set up TranD conda env**

1) download copy of TranD scripts from github (https://github.com/McIntyre-Lab/TranD)

2) create a conda env for running TranD:

$ conda create -p /path/to/conda\_envs/TranD

3) install TranD package in TranD conda env

activate TranD conda env: $ conda activate /path/to/conda\_envs/TranD

install dependencies: $ conda install python bedtools pysam

install TranD package:

cd into the TranD directory containing ‘setup.py’:

$ cd /to/location/of/TranD/source

$ pip install -e .

4) deactivate the TranD conda env: $ conda deactivate

5) Using the TranD conda env: Note: you don’t need to activate a specific conda environment to use that conda environment to run tools. To use a conda installed environment, add the path to it’s ‘bin’ directory to the PATH variable in your shell:

$ export PATH=/PATH/TO/SOME/CONDAENV/bin:$PATH

**TranD Usage**

usage: trand [-h] [-o OUTDIR] [-l LOG\_FILE] [-c] [--consolidate] [--consolPrefix CONSOL\_PREFIX] [-e {pairwise,gene}] [-k] [-p {all,both,first,second}] [-n CPU] [-1 NAME1][-2 NAME2] [-f] [-s] [-i] [-v]

input\_file [input\_file ...]

Perform transcript distance, complexity and transcriptome comparison analyses.

positional arguments:

input\_file, One or two input GTF file(s).

optional arguments:

-h, --help,

Show this help message and exit

-o OUTDIR, --outdir OUTDIR,

Output directory. If directory does not exist, it will be created.

Default:current directory.

-l LOG\_FILE, --logfile LOG\_FILE,

Log file name for logging processing events to file.

--consolidate,

Used with 1 GTF input file,

Consolidate transcripts with identical junctions prior to complexity calculations (remove 5'/3' transcript end variation in redundantly spliced transcripts), events and summary plotting.

Default: No consolidation.

--consolPrefix CONSOL\_PREFIX,

Used with 1 GTF input file,

Requires ‘consolidate’ flag. Specify the prefix to use for consolidated transcript\_id values. Prefix must be alphanumeric, can only include "\_" special character and not contain any spaces.

Default: tr

-c, --complexityOnly,

used with 1 or 2 GTF input files,

Output only complexity measures. If used in the presence of the '--consolidate' flag, complexity is calculated on the consolidated GTF(s).

Default: Perform all analyses and comparisons including complexity calculations

-e {pairwise,gene}, --ea {pairwise,gene},

Specify type of within gene transcript comparison:

‘pairwise’ – used with 1 or 2 GTF input files,

to compare pairs of transcripts within a gene,

‘gene’ – used with 1 GTF input file,

compare all transcripts within a gene

Default: pairwise

-k, --keepir,

used with 1 GTF input file,

Keep transcripts with Intron Retention(s) when generating transcript events.

Default: remove

-p {all,both,first,second}, --pairs {all,both,first,second},

Used with 2 GTF input files,

The TranD metrics can be output for all transcript pairs in both GTF files or for a subset of transcript pairs using the following options:

‘both’ – TranD metrics for the minimum pairs in both GTF files,

‘first’ – TranD metrics for the minimum pairs in the first GTF file,

‘second’ – TranD metrics for the minimum pairs in the second GTF file,

‘all’ – TranD metrics for all transcript pairs in both GTF files,

Default: ‘both’

-1 NAME1, --name1 NAME1,

Used with 2 GTF input files,

User-specified name to be used for labeling output files related to first GTF file. Must be alphanumeric, can only include ‘\_’ special character and not contain any spaces.

Default: d1

-2 NAME2, --name2 NAME2,

Used with 2 GTF input files,

User-specified name to be used for labeling output files related to second GTF file. Must be alphanumeric, can only include ‘\_’ special character and not contain any spaces.

Default: d2

-n CPU, --cpu CPU,

Number of CPUs to use for parallelization (default: 1).

-f, --force,

Force overwrite of existing output directory and files within.

-s, --skipplots,

Skip generation of all plots.

-i, --skip-intermediate,

Skip output of intermediate files (such as junction and exon region/fragment files).

-v, --verbose,

Verbose output

**Overview**

TranD has 2 overarching modes; 1 GTF or 2GTF.

In 1-GTF mode, structural differences within a single transcriptome are described and quantified directly on the input GTF file or on a 5’/3’ consolidated GTF file generated by tranD.

**1 GTF Input**

TranD can be used to ‘consolidate’ an individual transcriptome GTF. Here, ‘consolidate’ means to combine, by gene, those transcripts that vary only in the length of the TSS or TTS (all other splice junctions are identical). The representative transcript that is returned contains the longest TSS and the longest TTS among the transcripts with identical splice junctions.

TranD generates complexity measures from the input GTF file or, if specified, from the consolidated GTF file. Gene level complexity measures can be calculated by comparing pairs of transcripts within a gene (‘pairwise’) or by comparing all transcripts within a gene (‘gene’). If ‘gene’ mode is selected then a CSV file containing all possible exon regions and exon fragments is created along with a summary plot describing the transcriptome.

Input

1. GTF file of transcriptome of interest. File must contain “exon” features with transcript\_id and gene\_id values indicated in attributes.
2. Include consolidate flag (--consolidate) to consolidate transcripts prior to generating complexity measures. If omitted, the tool will not consolidate transcripts in the input GTF file. Complexity measure and events will be generated on the input GTF file directly.
3. Include complexity only flag (--complexityOnly) to generate complexity measures and exit before generating events.
4. A user-specified prefix (--consolPrefix) to be added to consolidated transcript\_id values (default: “tr”).
5. Full path to output directory. If the directory does not exist it will be created. To overwrite output files in a preexisting directory, use the –force/-f option.

Example command line usage

A) calculate complexity measures only (do not generate events) on a GTF file without consolidation, overwrite any existing output files:

$ trand GTF.gtf --complexityOnly --outdir /path/to/output –force

output files:

1. Box plots of complexity metrics (*complexity\_plots.png*)

* 1. number of transcripts per gene
  2. number of unique exons per gene
  3. number of exons per transcript

2. Figure legend for complexity metrics box plots (*complexity\_plots.rtf*).

3. Full table of complexity counts for the transcriptome (*transcriptome\_complexity\_counts.csv*). Contains the following columns:

num\_transcript

num\_gene

num\_exon

num\_uniqExon

min\_transcriptPerGene

min\_exonPerTranscript

min\_exonPerGene

q1\_transcriptPerGene

q1\_exonPerTranscript

q1\_exonPerGene

median\_transcriptPerGene

median\_exonPerTranscript

median\_exonPerGene

q3\_transcriptPerGene

q3\_exonPerTranscript

q3\_exonPerGene

max\_transcriptPerGene

max\_exonPerTranscript

max\_exonPerGene

mean\_transcriptPerGene

mean\_exonPerTranscript

mean\_exonPerGene

std\_transcriptPerGene

std\_exonPerTranscript

std\_exonPerGene

B) calculate complexity measures only (do not generate events) on a GTF file after consolidation, overwrite any existing output files:

$ trand GTF.gtf --consolidate --complexityOnly --outdir /path/to/output –force

output files:

In addition to the above files (*complexity\_plots.png, complexity\_plots.rtf, transcriptome\_complexity\_counts.csv*) the following GTF and CSV file is generated:

1. A consolidated GTF file (‘*consolidated\_transcriptome.gtf*’). Transcript\_id values contain the user-specified prefix (or default “tr” prefix). Each transcript\_id value is in the format [*prefix*]\_[*gene\_id*]\_[*number*], where *number* is a by gene enumeration value set after sorting by gene and transcript length. (NOTE: the consolidated GTF file contains only “exon” features).

2. Key file (*transcript\_id\_2\_consolidation\_id.csv*) connecting each transcript\_id value in the input GTF to the transcript\_id value in the consolidated output GTF.

a. gene\_id

b. transcript\_id: transcript\_id in the starting GTF file

c. consolidation\_transcript\_id, transcript\_id in the output consolidation GTF file

C) calculate complexity measures on consolidated GTF file, generate event file by comparing all transcripts for a gene and create a transcriptome summary plot (note that if you want to keep intron retention events add the --keepir):

$ trand GTF.gtf --consolidate --consolPrefix tr --ea gene --outdir /path/to/output –force

output files:

In addition to the above files (*complexity\_plots.png, complexity\_plots.rtf, transcriptome\_complexity\_counts.csv*, *consolidated\_transcriptome.gtf, transcript\_id\_2\_consolidation\_id.csv*) the following CSV files are generated:

1.

D) calculate complexity measures on consolidated GTF file, generate event file by a pairwise comparison of transcripts for a gene (keeping intron retention events) and calculate distances:

$ trand GTF.gtf --consolidate --consolPrefix tr --ea pairwise --keepir --outdir /path/to/output --force

Output Files

1. A consolidated GTF file (‘consolidated\_transcriptome.gtf’). Transcript\_id values contain the user-specified prefix (or default “tr” prefix). Each transcript\_id value is in the format [*prefix*]\_[*gene\_id*]\_[*number*], where *number* is a by gene enumeration value set after sorting by gene and transcript length. (NOTE: the consolidated GTF file contains only “exon” features).
2. Key file (*transcript\_id\_2\_consolidation\_id.csv*) connecting each transcript\_id value in the input GTF to the transcript\_id value in the consolidated output GTF.
   1. gene\_id
   2. transcript\_id: transcript\_id in the starting GTF file
   3. consolidation\_transcript\_id, transcript\_id in the output consolidation GTF file
3. Full table of complexity counts for the transcriptome (*transcriptome\_complexity\_counts.csv*). Contains the following columns:

num\_transcript

num\_gene

num\_exon

num\_uniqExon

min\_transcriptPerGene

min\_exonPerTranscript

min\_exonPerGene

q1\_transcriptPerGene

q1\_exonPerTranscript

q1\_exonPerGene

median\_transcriptPerGene

median\_exonPerTranscript

median\_exonPerGene

q3\_transcriptPerGene

q3\_exonPerTranscript

q3\_exonPerGene

max\_transcriptPerGene

max\_exonPerTranscript

max\_exonPerGene

mean\_transcriptPerGene

mean\_exonPerTranscript

mean\_exonPerGene

std\_transcriptPerGene

std\_exonPerTranscript

std\_exonPerGene

1. Box plots of complexity metrics (*complexity\_plots.png*)
   1. number of transcripts per gene
   2. number of unique exons per gene
   3. number of exons per transcript
2. Figure legend for complexity metrics box plots (*complexity\_plots.rtf*).

Example usage to consolidate a GTF file, comparing all transcripts for a gene, generate events and output figure summarizing transcriptome:

$ trand /input/GTF.gtf --consolidate --ea gene --outdir /path/to/output --force

Comparison of Two Transcriptomes

Overview

Compare two transcriptomes that are from the same genomic coordinates (both sets of transcript coordinates map to the same genome). Transcriptomes must have shared gene\_id values for transcripts to be compared within each shared gene (See X for how to associate gene\_id values with mapped transcripts using SQANTI3 QC).

Input

1. Two GTF files corresponding to the transcriptomes of interest. Files must contain “exon” features with transcript\_id and gene\_id values indicated in attributes.
2. Names associated with each transcriptome to be used in output tables and plots (default: “d1” and “d2”). Names must be alphanumeric with no spaces or special characters other than “\_”.
3. Full path to output directory, if it does not exist, it will automatically be created. The directory can already exist, and the output files will be replaced if present in directory and “--force/-f” option is set.
4. Output category option allows user to select the output table of transcript pairs to only be minimum pairs from the “first” transcriptome, only minimum pairs from the “second” transcriptome, minimum pairs for “both” transcriptomes (minimum pair in either), or output “all” pairwise transcript pairs regardless of minimum pair classification. (Default: “both”)
5. Number of CPU to use for parallelization (default: 1, no parallelization). TranD splits by gene to each CPU so if you have many genes with lower numbers of transcripts then increasing this value will help with speed (genes with higher numbers of transcripts may not benefit as well).
6. *Optional*: Indicator to skip output of intermediate files (“--skip-intermediate/-i”) and only output the minimum distance output file and all plots.

Output

1. Output CSV file (called *pairwise\_transcript\_distance.csv* if output category is “all”, or *minimum\_pairwise\_transcript\_distance.csv* otherwise) which includes the transcript distance and minimum distance output for all pairwise transcripts pairs (“all”), all minimum pairs of the first transcriptome to the second transcriptome (“first”), all minimum pairs of the second transcriptome to the frist transcriptome (“second”), or the minimum pairs for either the first transcriptome and/or the second transcriptome (“both”). The distance columns are described in *transcript\_distance\_column\_descriptions.xlsx* (first 56 columns of output file). The minimum distance columns are described in *minimum\_distance\_column\_descriptions.xlsx* (last 36 columns of output file).
2. Box plots of complexity metrics (*[name]\_complexity\_plots.png*) and automatically generated figure legend (*[name]\_complexity\_plots.rtf*) for each transcriptome, where *[name]* represents the names given to each transcriptome (or “d1” and “d2” as default).
3. Various plots described in TranD manuscript and automatically generated figure legends for each (*gene\_avg\_nt\_diff\_pairs.png/.rtf, gene\_avg\_nt\_diff\_pairs\_zoomIn.png/.rtf, recip\_min\_pair\_AS\_upset\_nt\_box.png/.rtf, transcript\_in\_gene\_split\_pie.png/.rtf, transcript\_in\_gene\_stackCount.png/.rtf, transcript\_in\_gene\_stack\_proportion.png/.rtf, transcript\_in\_gene\_upset.png/.rtf, xcrpt\_gene\_AS\_upset\_nt\_box.png/.rtf*).
4. If intermediate files are output (no “-i”), CSV file of pairwise events generated for each pair is output (*event\_analysis.csv*).

Example command line for comparing two transcriptomes:

trand \

/path/to/reads.gtf \ # GTF of first transcriptome

/path/to/ref.gtf \ # GTF of second transcriptome

-o /path/to/outdir \ # TranD output directory

-1 name1 \ # Name for first transcriptome (default: “d1”)

-2 name2 \ # Name for second transcriptome (default: “d2”)

-p both \ # Output option for table of transcript pairs

-i \ # Skip intermediate output files

-f \ # Force output to overwrite if running again

-n 8 # Number of CPU (default: 1)