TRANSECT Manual 24.04

1. Background

TRANSECT works by defining two groups (strata, plural for stratum) within a cohort based solely on the expression of a gene or a gene set of interest and subsequently compares the stratum, one against the other, for global expression changes and functional differences. TRANSECT outputs descriptive statistics about the gene/s of interest, the products of the stratification process, the differential expression results and subsequent enrichment outcomes. The application uses publicly available large cohort datasets and simply requires the user to choose at a minimum

1. the **cohort database** containing participant IDs and gene expression measurements
2. a **gene (or multiple genes)** of interest whose expression levels are used to rank participants in the cohort database
3. an **integer percentile** value used on the expression or ranking measurements as a threshold to partition the cohort into low and high stratum for subsequent comparisons

2. Getting started

2.1 Installation using Conda

(*10 - 15 minutes on an average PC*)

Making use of a Conda environment for the sizable number of prerequisite modules and dependencies needed by TRANSECT is recommended for most use cases. It is not only easier to achieve but cleaner, simpler to manage and way quicker than the native install

1. Start by cloning the repository using the git command to a suitable location on your device

```
$ git clone https://github.com/twobeers75/TRANSECT.git
```

   Alternatively TRANSECT code and executable files can be downloaded from GitHub at https://github.com/twobeers75/TRANSECT. Click on the green "Code" button followed by "Download ZIP" (note the download location). Find the downloaded ZIP file and move it to an appropriate location if required, before extracting the contents and renaming the folder

```
$ unzip TRANSECT-main.zip
$ mv TRANSECT-main TRANSECT
```

2. Install Conda on your system (version > 24.1.0). You can skip this step if you already have it. There are many wikis on how to install Conda for Ubuntu, here is just one. Please consult the Conda documentation relevant to your operating system.

```
### follow the instructions outlined in the link above which should look
something like this for Linux
$ mkdir -p ~/miniconda3
$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
-O ~/miniconda3/miniconda.sh
$ bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
$ rm -rf ~/miniconda3/miniconda.sh

### don't forget to initialize the bash shell. Mac users need to check their
default shell and change the following command appropriately
$ ~/miniconda3/bin/conda init bash

### Afterwards, you will be asked to restart your terminal whereby you should
see (base) at the prompt. Ignore it for now.
```

3. Next, we create the TRANSECT Conda environment. Here we will run an installation script that automates the creation of the TRANSECT environment, with all the tools and dependencies required to run TRANSECT. The scripts required for this can be found in the INSTALL/ subdirectory of the TRANSECT folder.

```
### change into the top directory of the downloaded folder (TRANSECT) and
navigate to the INSTALL folder
$ cd <path to>/TRANSECT/INSTALL

### First, run the conda install script
$ ./TRANSECT_conda_install.sh

### Next, upon successful completion of the previous step, activate the newly
created environment
$ conda activate TRANSECT

### Finally, whilst still within the TRANSECT/INSTALL directory and in the
TRANSECT environment run the post installation script to complete the setup
$ ./TRANSECT_post_conda_install.sh

### You now need to reactivate the TRANSECT environment to apply the changes.
$ conda deactivate
$ conda activate TRANSECT
```

And that's it! You should now have all the necessary applications and dependencies in the TRANSECT environment to run this application. Please note, just like any virtual environment you are required to activate the TRANSECT environment in order to use the application. You can deactivate at will when not in use.

A few extra very useful commands for those not accustomed to Conda environments

```
### By default, Conda auto-activates the "base" default environment so each time
you open a terminal you will automatically be in the (base) environment. I prefer
not to have this happen. To disable, run the following commands.
#First, deactivate any environment until there is no (XXX) at the beginning of
the prompt, then turn off auto_activate_base
$ conda deactivate
$ conda config --set auto_activate_base false

### To check what environments you have on your system
$ conda env list
### You should see "base" and "TRANSECT" at the very least

### To activate TRANSECT at any time
$ conda activate TRANSECT
### Once you have finished, to deactivate the TRANSECT environment
$ conda deactivate

### To remove/uninstall the TRANSECT environment
conda remove -n TRANSECT -all
```

More information about managing Conda environment can be found here

2.2 Native installation on Ubuntu

(*30 - 45 minutes or longer on an average PC*)

NOTE: This is not the recommended installation procedure. TRANSECT requires and depends on numerous packages and applications. These take some time to install natively if not already present. A fresh install on a vanilla Ubuntu 22.04 can take 30-45mins depending on the PC and network speeds.

1. To start, clone the repo

```
$ git clone https://github.com/twobeers75/TRANSECT.git
```

   Alternatively TRANSECT code and executable files can be downloaded from GitHub at https://github.com/twobeers75/TRANSECT. Click on the green "Code" button followed by "Download ZIP" (note the download location). Find the downloaded ZIP file and move it to an appropriate location if required, before extracting the contents and renaming the folder

```
$ unzip TRANSECT-main.zip
$ mv TRANSECT-main TRANSECT
```

2. Install python3 pip, java if required, and other TRANSECT dependencies
   (approx. 1-2min)

```
### change into the top directory of the downloaded folder (TRANSECT)
$ cd <path to>/TRANSECT

### install pip and other deb requirements
$ sudo apt install python3-pip default-jre libfontconfig1-dev libcurl4-
openssl-dev libssl-dev libxml2-dev libharfbuzz-dev libfribidi-dev
libfreetype6-dev libpng-dev libtiff5-dev libjpeg-dev pandoc

### install python modules
$ python3 -m pip install -r pip_requirements.txt
```

3. Install R, the "pacman" package and Bioconductor specific packages. You can skip this step if you already have R. There are many wikis on how to install R on Ubuntu, here is just one (specifically for Ubuntu 22.04) *(approx. 1min)*

```
# follow the instructions outlined in the link above which should look
something like this
$ wget -qO- https://cloud.r-project.org/bin/linux/ubuntu/marutter_pubkey.asc
| sudo gpg --dearmor -o /usr/share/keyrings/r-project.gpg
$ echo "deb [signed-by=/usr/share/keyrings/r-project.gpg] https://cloud.r-
project.org/bin/linux/ubuntu jammy-cran40/" | sudo $ tee -a
/etc/apt/sources.list.d/r-project.list
$ sudo apt update
$ sudo apt install r-base
```

4. Start R from the terminal and install pacman and devtools. Follow the prompts and choose (if asked) to install these packages into a personal library.

   Once you enter the R shell you should see printed out in the terminal a number of lines about the R version and licenses followed by a ">" symbol. I have used this symbol below to indicate that you need to be in the R shell to run these commands but, you can't copy the ">" symbol too. It won't work. *(approx. 25mins)

```
### start R
R
> install.packages(c("pacman","devtools"))
# Note: maybe wise here to go get a coffee as the previous command takes
quite some time to finish! (approx. 15mins)

### whilst still in the R environment, load devtools and install rlogging
> library("devtools")
> install_github("https://github.com/mjkallen/rlogging.git")

### also install required Bioconductor packages
> if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
> BiocManager::install(version = "3.19")
> BiocManager::install(c("edgeR","Glimma","DEFormats"))
# Note: probably time for another coffee. Sorry! (approx. 10mins)

### once successfully completed you can quit R, no need to save the
workspace.
# No more coffee for you today ;-)
>  q()
```

NOTE: TRANSECT requires many additional R packages however these are all installed on demand the first time (and only the first time) you run each one of the different TRANSECT commands after a nativ install. Please keep this in mind on your first run as it will take substantially longer compared to all subsequent runs.

3. Basic demonstration

TRANSECT has two main operations; **Prepare** and **Analyse**.

In order for TRANSECT to function, it first requires a cohort dataset to work on. This is retrieved and formatted appropriately by the **Prepare** scripts. Subsequently, TRANSECT can run analyses on the downloaded data using the **Analyse** scripts

Example commands to investigate ZEB1 using the RECOUNT3 TCGA PRAD cohort;

```
### First, if not already make sure to activate the TRANSECT environment
conda activate TRANSECT

### Next Download the RECOUNT3 PRAD data (approx. 3mins)
# run the RECOUNT3 prepare script for TCGA-PRAD (the resulting files can be found
in <path to>/TRANSECT/data/RECOUNT3/PRAD)
R3_prepare_directories.sh -p PRAD
# NOTE: you only need to do this one time per cohort dataset. Once you have it
you don't need to download it again!

### Finally, Run the TRANSECT analysis
# TRANSECT saves output in the current working folder so best to create a new
folder specifically for each run
cd <path to>/TRANSECT/output/RECOUNT3
mkdir -p ZEB1_PRAD_test
cd ZEB1_PRAD_test

# Now, run the RECOUNT3 analyse script using the PRAD data we just retreived,
investigating the gene ZEB1, partitioning the participants using a percentile
threshold of 5, with all outputs.
R3_analyse_GOI.sh -p PRAD -g ZEB1 -s mRNA -t 5 -a

### Done! Explore the outputs. Try swapping ZEB1 for your favourite gene
(remember, new folder for each run).
```

4. Stratification modes

The basic premise of TRANSECT is to stratify individuals from large cohort transcriptomic data into defined groups called strata (plural for stratum) based on singular gene expression or composite gene expression sets. The stratified participant strata are subsequently compared one to the other in order to assess global expression changes and functional differences.

4.1 Singular gene analysis

Singular gene stratification is simply the division of individuals within a cohort population into distinct strata based solely on the expression of one single gene. In the current version of TRANSECT, individuals with expression levels at or near both ends of the physiological limits for the gene of interest are grouped separately and subsequently compared.

4.2 Composite gene analysis - Additive mode

Composite analyses use information from multiple genes simultaneously to divide individuals within a cohort population into distinct strata. The additive mode of TRANSECT uses expression information from multiple genes (2 – 5 genes in the current implementation) to rank individuals expressing each of the component genes at near to physiological extreme and separate them into low and high strata. This is achieved by computing the average of rank positions for all component genes for each participant and using the metric to position each individual within the cohort in order. Once this is achieved, individuals with extreme high average rank positions can be grouped and compared to individuals with extreme low rank positions.

4.3 Composite gene analysis - Ratio mode

In the same manner as the additive mode described above, the ratio mode also considers information from multiple genes simultaneously to partition individuals within a cohort population into distinct strata. The ratio mode uses expression information from strictly two genes to rank individuals, looking for participants that show converse expression of the two. In order to achieve this, TRANSECT calculates a simple log-ratio statistic between these 2 genes for each patient. Extremely low ratio scores will demarcate participants where geneA >>> geneB and vice versa. Again, individuals at both extremes are grouped and compared.

4.4 Multimodal analysis

In select large cohort studies there exist measurements derived from multiple omics for the same individual at the same or similar timepoints. For example, the TCGA study consists of RNA (mRNA, miRNA), and DNA (methylation, mutation, and copy number) data in addition to the associated global proteomics data generated by the Clinical Proteomic Tumor Analysis Consortium (CPTAC). TRANSECT has the facility to survey changes in one omics data type based on the stratification of individuals using matched data from another omics. As in the use cases above, individuals at each extreme are grouped and compared.

5. Prepare commands and options

Prepare is a process that retrieves the raw data from online repositories and prepares it (if required) for analysis. TRANSECT comes bundled with three different prepare scripts, one each for RECOUNT3, GTEx and GDC-TCGA data.

All downloaded and formatted data is stored by default in the TRANSECT/data/<RECOUNT3|GTEx|GDC>/ subdirectory in individual folders named by tissue/cancer abbreviation. For example, the RECOUNT3 PRAD data downloaded in the basic demonstration of this manual is stored in /TRANSECT/data/RECOUNT3/PRAD/

5.1 RECOUNT3

Retrieve and prepare RECOUNT3 RNA-seq data for in-house custom analyses.

USAGE:

```
$ R3_prepare_directories.sh [-h] -p <RECOUNT3 Project ID>
```

PARAMETERS:

-h Show help text

-p RECOUNT3 project id: needs to be valid RECOUNT3 project id (ie. BRCA for TCGA data OR BREAST for GTEx). **Required**

You can retrieve and prepare more than one RECOUNT dataset by using a bash for loop like this;

```
$ for r3_code in COAD BREAST LAML; do R3_prepare_directories.sh -p $r3_code ;
done
```

5.2 GTEx

Retrieve and prepare GTEx RNA-seq data for in-house custom analyses. Unlike RECOUNT3 and GDC data retrieval, GTEx data for all tissue types are retrieved in a single file. Subsequently, this is separated into tissue specific datasets using the information in the metadata file.

USAGE:

```
$ GTEx_prepare_directories.sh [-h] [-a -c -t]
```

PARAMETERS:

-h Show help text

-a retrieve all expression data (mRNA counts and TPMs), **Required** for the proper functioning of TRANSECT

-c retrieve only mRNA counts

-t retrieve only mRNA TPMs

5.3 GDC

Retrieve and prepare TCGA RNA-seq data for in-house custom analyses.

USAGE:

```
$ GDC_TCGA_prepare_directories.sh [-h] -p <TCGA Project ID> [-a -c -r -R -k -n]
```

PARAMETERS:

-h Show help text

-p TCGA project id: needs to be valid TCGA project id as used by GDC (ie. TCGA-BRCA). **Required**

-a retrieve all expression data (mRNA counts and TPMs as well as miR and isomiR RPMs)

-c retrieve only mRNA counts

-r retrieve only miR RPMs

-R retrieve only isomiR RPMs

-k keep all data (Default: False)

-n data is not from TCGA study (Default: False)

To retrieve and prepare more than one TCGA cancer dataset use a bash for loop like this;

```
$ for tcga_code in TCGA-COAD TCGA-SARC TCGA-LAML; do
GDC_TCGA_prepare_directories.sh -p $tcga_code; done
```

To retrieve and prepare all TCGA cancer datasets you can loop through all lines in GDC_API/TCGA_Study_Abbreviations.tsv (WARNING: this requires lots of time, network and disc space)

```
$ while read tcga_code; do GDC_TCGA_prepare_directories.sh -p $tcga_code; done <
<(cut -f1 GDC_API/TCGA_Study_Abbreviations.tsv | tail -n +2)
```

Please be aware that some of these collections are large and require substantial disk space. They can take a considerable amount of time to download and process. For example, downloading and processing GDC TCGA-BRCA takes just over 30 minutes (using a high speed network connection and an up to date workstation) and requires more than 14GB of disk space (most of which can and by default is, deleted afterwards). In comparison, GDC TCGA-LAML takes less than 5 minutes to retrieve and less than 2GB of disc space.

In addition, the GDC prepare script often fails when downloading large datasets. This is caused by network connectivity issues (tested only in Australia) with the GDC repository. If you experience issues, delete the relevant dataset and retry the prepare command.

6. Analysis commands and options

Analyse is a process that uses the prepared public data from above, conducts the stratified differential expression and produces all the outputs. Like with the prepare operations, TRANSECT comes bundled with three analyse scripts, one each for RECOUNT3, GTEx and GDC-TCGA.

Unlike the prepare operations, the output from these calls is saved in the current working directory and therefore it is recommended to create a descriptively named folder for each of your analyses. TRANSECT comes with an preinstalled output folder containing subdirectories (TRANSECT/output/<RECOUNT3|GTEx|GDC>/) however, you may choose any working directory at your discretion. Keep in mind that if TRANSECT output exists in the current working directory, it will be overwritten.

For each script, composite analyses can be run using the plus charater (+) for additive combinations or by using the modulus character (%) for ratio. The two special characters are used between gene names like so. Additive example: ESR1+PGR+ERBB2 or Ratio example: ESRP1%ZEB1

6.1 RECOUNT3

Differential expression analysis of RECOUNT3 data stratified into high and low groups by gene of interest
Please run this wrapper script in the directory of the desired output location

USAGE:

```
$ R3_analyse_GOI.sh [-h] -p <RECOUNT3 projectID> -g <GOI> -s <StratifyBy> -t
<Percentile> -e -S -a -c -d
```

PARAMETERS:

-h Show help text

-p RECOUNT3 tissue id: needs to be valid RECOUNT3 tissue id as at RECOUNT3 (ie. BRCA for TCGA or BREAST for GTEx). **Required**

-g Gene of interest: needs to be a valid HGNC symbol (ie. ZEB1). **Required**
-s Stratify by molecule: Must match -g and can only be mRNA at present. **Required**
-t Percentile: startify data into top and bottom x percentile (valid x between 2 and 25). **Required**
-e Enrichment analyses: Run GSEA on DE results (Default: Only run WebGestalt)
-S Switch pairwise comparison: find genes DE in low group compared to high group (Default: high compared to low)
-a Do all analyses
-c Do correlation analysis only
-d Do differential expression analysis only

6.2 GTEx

Differential expression analysis of GTEx data stratified into high and low groups by gene of interest
Please run this wrapper script in the directory of the desired output location

USAGE:

```
$ GTEx_analyse_GOI.sh [-h] -p <GTExTissueID> -g <GOI> -s <StratifyBy> -t
<Percentile> -e -S -a -c -d
```

PARAMETERS:
-h Show help text
-p GTEx tissue id: needs to be valid GTEx tissue id as at GTEx (ie. Breast). **Required**
-g Gene of interest: needs to be a valid HGNC symbol (ie. ZEB1). **Required**
-s Stratify by molecule: Must match -g and can only be mRNA at present. **Required**
-t Percentile: startify data into top and bottom x percentil (valid x between 2 and 25). **Required**
-e Enrichment analyses: Run GSEA on DE results (Default: Only run WebGestalt)
-S Switch pairwise comparison: find genes DE in low group compared to high group (Default: high compared to low)
-a Do all analyses
-c Do correlation analysis only
-d Do differential expression analysis only

6.3 GDC

Differential expression analysis of TCGA data stratified into high and low groups by gene of interest
Please run this wrapper script in the directory of the desired output location

USAGE:

```
$ GDC_TCGA_analyse_GOI.sh [-h] -p <TCGAProjectID> -g <GOI> -s <StratifyBy> -t
<Percentile> -e -S -a -c -d
```

PARAMETERS:
-h Show this help text
-p TCGA project id: needs to be valid TCGA project id as at the GDC (ie. TCGA-BRCA). **Required**
-g Gene of interest: needs to be a valid HGNC symbol (ie. ZEB1). **Required**
-s Stratify by molecule: must match -g and can only be one of (mRNA or miRNA). **Required**

-t Percentile: startify data into top and bottom x percentile (valid x between 2 and 25).

Required

-e Enrichment analyses: Run GSEA on DE results (Default: Only run WebGestalt)

-S Switch pairwise comparison: find genes DE in low group compared to high group (Default: high compared to low)

-a Do all analyses

-c Do correlation analysis only

-d Do differential expression analysis only

7. Output

TRANSECT takes in a cohort dataset and processes the data as follows.

1. First, TRANSECT partition the data by the expression of a gene/s of interest into low and high strata
2. Subsequently, TRANSECT compares the resulting strata, one to the other, to identify differentially expressed genes
3. And finally, TRANSECT uses the results from the DE analysis to run functional annotation and enrichment analyses

The outputs from TRANSECT are likewise grouped into 3 categories and returned in three folders in the working directory from where the program is executed

7.1 01-Stratification

The stratification process produces 2 tables, and 3 plots.

1. GOI_exp_raw_OG.tsv contain the raw original expression (TPM) data for all gene/s of interest
2. GOI_exp_with_strat.tsv contains the same data sorted with additional columns relating to the participants ranking score, percentiles and quantile values.
3. TPM_histogram.html or TPM_Boxplot_Sina.html or TPM_Scatter.html, all which plot data from the two tables above differently depending on the chosen TRANSECT mode in an attempt to describe the distribution of gene expression across the cohort participants
4. TPM_N-T_boxplot.html which shows the distribution of expression partitioned by disease state when available
5. TPM_strat_boxplot.html which plots the low and high strata participants resulting from the stratification process

7.2 02-DE

The DE analysis produces many tables and plots most easily described as follows.

1. DE Setup – design.tsv and gene_raw_expression_data_cpm.csv
2. DE QC – bcv and mean_var.png plots as well as the MDS-Plot.html in the glimma-plots folder
3. Normalised expression tables - gene_normalised_expression_data_cpm.csv (also in log form)
4. DE result tables - High_Vs_Low _de_sigFC.csv and top_tags.csv
5. DE result plots - High_Vs_Low_volcano.png and High_Vs_Low_heatmap.png as well as an interactive version of the volcano plot in the glimma-plots folder

6. The glimma-plots folder containing the interactive web plots and associated data

7.3 03-Enrichment

The 2 enrichment analyses result in the production of two folders each with a separate collection of tables and plots.

1. GSEA

   When selected, this folder contains the output folders from running GSEA against the Hallmark as well as the Curated MSigDB collections respectively. Within each folder, users can open the index.html file to access and interact with the results in a web browser. In addition, the results are summarised and provided in tabular form (.csv) as well as interactive form (.html)

   GSEA input data – 3 text files used for the GSEA analysis are saved in the top-level folder. The default GSEA method used by TRANSECT is the pre-ranked method. Input for this analysis can be found in the .rnk file. Provided but not used by TRANSECT are alternate GSEA input files (.cls and .txt).

2. WebGestalt

   The ORA results are presented in six folders; two each for disease, gene ontology and pathway enrichment, for up and down regulated genes separately (when available). Within each folder, users can open the .html file to access and interact with the results in a web browser.

8. Precautions

8.1 Cohort size

TRANSECT requires large numbers of participants in the cohort data sets to adequately achieve appropriate stratification and grouping. Ideally, individual members of each stratum derived from the stratification process will share highly similar attributes or characteristics (here, gene expression levels). Cohort data sets with low participant numbers are unlikely to possess the required random sampling of a population to achieve defined stratum containing members with shared characteristics and may force the allocation of members with different characteristics into the same stratum.

8.2 Bulk RNA-seq heterogeneity

The heterogeneity of cell types within bulk tissue samples which are present in these cohort data sets can lead to misleading observations if not carefully considered

9. Publication

Publication details to come (hopefully!)