StrokeofGenus Vignette

Kurt Weir

2024-02-11

Contents

Introduction

StrokeofGenus is a pipeline to identify distinct transcriptomic states and conservation of gene expression patterns across studies, tissues, and species in bulk RNA-seq.

StrokeofGenus identifies orthology across species without the need for a reference genome to enable comparison of gene expression in non-model organisms and uses machine learning for the discovery of data structure and comparison across datasets. It is able to identify transcriptomically distinct (and indistinct) states and cross-study conservation of gene expression programs. By identifying conserved gene expression programs across species, we demonstrate the presence of shared molecular mechanisms.

This vignette will walk you through:

- The software necessary to support this pipeline
- The folder structure before running StrokeofGenus
- The first step of running StrokeofGenus which covers de novo transcript assembly, gene expression quantification, dataset structure identification, translation, and the production of additional files for orthology identification
- The folder structure necessary to run orthology identification
- The second step of running StrokeofGenus which covers orthology identification
- The third step of running StrokeofGenus which covers the identifcation of conserved gene expression between any two datasets for which orthology has been identified

Before running any of these examples, it is highly recommended that the user establishes a new empty directory to consolidate files. Many of StrokeofGenus' functions create multiple output files.

Installation

```
mkdir path/to/StrokeofGenus

cd StrokeofGenus

git clone https://github.com/vbusa1/StrokeofGenus_manuscript
```

Software Dependencies

Create a "yard" directory to store all dependencies. Add all dependency directories to your PATH after installation. StrokeofGenus will use the full path to the yard directory for dependencies other than R, bowtie2, and Cuda.

Required dependencies include:

R

StrokeofGenus was developed using version 4.0.2 with the following libraries:

- CoGAPS v3.10.0
- projectR v1.6.0
- plyr v1.8.6
- dplyr v1.0.7
- seqinr v4.2-8
- phylotools v0.2.2

- tidyr v1.1.4
- Matrix v1.3-4
- stringr v1.4.0
- pheatmap v1.0.12
- RColorBrewer v1.1-2
- SparseM v1.81

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install(c("CoGAPS", "projectR", "plyr",
                       "dplyr", "seqinr", "phylotools",
                       "tidyr", "Matrix", "stringr",
                       "pheatmap", "RColorBrewer", "SparseM"))
```

Trinity

StrokeofGenus was developed using the Trinity Singularity image, which is the preferred way to use Trinity. If your system does not already have Singularity installed, Singularity can be downloaded from sylabs

```
singularity pull --name trinityrnaseq docker://trinityrnaseq/trinityrnaseq
```

StrokeofGenus was developed using v2.13.1.

TransDecoder

Download instructions for TransDecoder are available on the transdecoder wiki

TransDecoder requires no compilation

StrokeofGenus was developed using version v5.5.0

OMAStandalone

OMAStandalone is available for download at omabrowser

StrokeofGenus does not require installation of OMAStandalone to run, only download.

StrokeofGenus was developed using version v2.5.0

Cuda

Follow download instructions at nvidia.com

StrokeofGenus was developed using version v11.1.0

bowtie2

Follow download instructions at the bowtie2 GitHub

StrokeofGenus was developed using version v2.4.1

1. Download the data

Download the fastq files for the studies you want to analyze and place them in the correct folder. For this tutorial, you can use the MCF7 dataset from Li et al.

```
#Downloading from enaBrowser:
mkDir /path/to/StrokeofGenus/MCF7/Data
cd path/to/StrokeofGenus/MCF7/Data

# The MCF7 dataset was sequenced paired end.
# This will download both fastq files for each sample

wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR841/008/SRR8413888/*
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR841/000/SRR8413890/*
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR841/004/SRR8413884/*
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR841/009/SRR8413889/*
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR841/001/SRR8413891/*
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR841/005/SRR8413885/*
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR841/006/SRR8413886/*
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR841/007/SRR8413887/*
wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/SRR841/002/SRR8413892/*
```

StrokeofGenus does not include functions for sequence QC or trimming. Although this is not strictly necessary, it is highly advised. We suggest fastQC, MultiQC, and Trimmomatic for these steps.

2. Step 1: expression quantification and identify gene pattern

The first step of StrokeofGenus takes the input fastq files and outputs gene expression quantification, dataset structure identification, and files necessary for orthology identification. Warning: This step can take hours or even days depending on how many reads are being used for de novo transcript assembly.

Step 1 consists of the following files, in order:

- `trinity_to_splice_gen.sh`
- `trinity_to_splice_gen_kmer.sh`
- `trinity_to_splice_gen_inchworm.sh`
- `trinity_to_splice_gen_chrysalis.sh`
- `trinity_to_splice_gen_phase2.sh`
- `trinity_to_splice_gen_transdecoder_longorf.sh`
- `trinity_to_splice_gen_transdecoder_predict.sh`
- `trinity_to_splice_gen_transdecoder_splice_gen.sh`
- `trinity_to_splice_gen_rsem_abundance.sh`
- `trinity_to_splice_gen_gwCoGAPS.sh`

Modify scripts for cluster submission

StrokeofGenus was developed for the SLURM submission manager system. Edit the header of each .sh file to your appropriate queue and resource needs.

```
#!/bin/bash -l

#SBATCH
#SBATCH --job-name=Tri_start_Job
#SBATCH --partition=bigmem
#SBATCH -t 00-00:03:00
#SBATCH -N 1
#SBATCH --ntasks-per-node=20
#SBATCH --mail-type=ALL
#SBATCH --mail-user=you@youremail.edu
```

Preparing inputs

Trinity requires a tab-separated text file listing the condition, replicate, and the path to the fastq file(s) for each sample. If the dataset is single-end, you will only list the path to one fastq file per sample. An example file `samples.txt` is available in the Github.

Variables, such as species, path to the "yard" directory housing dependencies, and path to the sample file, are set in `trinity_to_splice_gen.sh`. Edit these variables in `trinity_to_splice_gen.sh`, shown below, then submit the script.

```
# Fill in these variables to your specific use-case

samples=path/to/StrokeofGenus/scripts/trinity_samples_MCF7.txt # file listing fastqs
output_folder=path/to/StrokeofGenus/ # will be appended to species below
scripts=path/to/StrokeofGenus/scripts/ # folder with StrokeofGenus scripts
yard=path/to/yard # directory containing dependencies
species=MCF7 # dataset species; best as a single word without underscores
CoGAPSmin=2 # minimum number of CoGAPS patterns
CoGAPSmax=10 # maximum number of CoGAPS patterns
```

When finished, this file will submit the next .sh file and so on.

Running only part of step 1

If you want to run a later script in Step 1 without rerunning an earlier script, enter the above as environmental variables and submit the relevant .sh file. If you don't want to run a later step, comment out the `sbatch` command at the bottom of the preceding .sh file (see order above, under 'Step 1: expression quantification and identify gene pattern').

Outputs

Step 1 of StrokeofGenus will produce the following outputs:

- `path/to/StrokeofGenus/MCF7/Trinity.Trinity.fasta`
- `path/to/StrokeofGenus/MCF7/Transdecoder/Trinity.Trinity.fasta.transdecoder.rename.fa`
- `path/to/StrokeofGenus/MCF7/Transdecoder/Trinity.Trinity.fasta.transdecoder.splice`
- `path/to/StrokeofGenus/MCF7/RSEM/MCF7_*/RSEM.genes.results`
- `path/to/StrokeofGenus/MCF7/CoGAPS/MCF7countsframe_filtered_log`
- `path/to/StrokeofGenus/MCF7/CoGAPS/MCF7_matslog.pdf`
- `path/to/StrokeofGenus/MCF7/CoGAPS/gwCoGAPS_testlog_MCF7_result_*.RData`
- `path/to/StrokeofGenus/MCF7/CoGAPS/MCF7_*patternslog.pdf`

In the CoGAPS folder, the `matslog.pdf` includes a boxplot of expression for each sample. To proceed, expression for all samples should be comparable. The `patternslog.pdf` files include heatmaps of the sample weights for each pattern identified by CoGAPS. Use these pdfs to determine how many signals are present in the dataset.

3. Establish folder structure

Appropriate folder structure is necessary for orthology identification. This step requires human decision-making and should only be run after all species that you want to compare have completed step 1 processing.

Set up the folder structure that OMAStandalone expects:

```
# Create an OMA folder.
mkdir path/to/StrokeofGenus/OMA/omaDir
# Within that folder, copy the parameters.drw file from the StrokeofGenus GitHub.
# It has been populated so that OMAStandalone installation is unnecessary and to avoid time-heavy steps
cp path/to/StrokeofGenus/scripts/parameters.drw path/to/StrokeofGenus/OMA/omaDir
# Create a DB folder within the OMA folder.
mkdir path/to/StrokeofGenus/OMA/omaDir/DB
```

Copy the pre-made OMAstandalone .fa and .splice files for all species to the `DB` folder. Always include `HUMAN` files so that gene expression for your datasets can be tied to known genes and the `YEAST.fa` file (there is no .splice file) as the outgroup for OMAStandalone.

```
cp path/to/StrokeofGenus/scripts/HUMAN.fa path/to/StrokeofGenus/OMA/omaDir/DB
cp path/to/StrokeofGenus/scripts/HUMAN.splice path/to/StrokeofGenus/OMA/omaDir/DB
cp path/to/StrokeofGenus/scripts/YEASO.fa path/to/StrokeofGenus/OMA/omaDir/DB
```

Copy `Trinity.rename.fasta` and `Trinity.rename.splice` files to `OMA/DB` folder. Rename the files to `species.fa` and `species.splice` for appropriate species. Ensure the species name is a single alphanumeric word without symbols.

```
cp path/to/StrokeofGenus/MCF7/Transdecoder/Trinity.Trinity.fasta.transdecoder.rename.fa \
    path/to/StrokeofGenus/OMA/omaDir/DB/MCF7.fa
cp path/to/StrokeofGenus/MCF7/Transdecoder/Trinity.Trinity.fasta.transdecoder.splice \
    path/to/StrokeofGenus/OMA/omaDir/DB/MCF7.splice
```

4. Step 2: identify orthologs

The second step of StrokeofGenus constructs an orthology matrix matching orthologs across all included species. It consists of three .sh files: `trinity_to_splice_gen_oma.sh` sets the variables; `trinity_to_splice_gen_oma_stage_1.sh` and `trinity_to_splice_gen_oma_stage_2.sh` split the OMA program into two steps with different resource requirements. Warning: This step can take hours or even days depending on how many species you are including in the analysis because OMAStandalone is built off pairwise comparisons and each species added leads to a greater-than-linear increase in computation time.

stage 1 submission

First, edit the header of each .sh file to your appropriate queue and resource needs. The first step of OMA is run as an array to speed up processing time.

Variables, such as path to the output folder and path to "yard" directory housing dependencies, are set in `trinity_to_splice_gen_oma.sh`. Edit `trinity_to_splice_gen_oma.sh` to set these variables and then submit the script. When you first submit oma, only `trinity_to_splice_gen_oma_stage_1.sh` will run.

```
# Fill in these variables to your specific use-case

output_folder=path/to/StrokeofGenus/
scripts=path/to/StrokeofGenus/scripts/
yard=path/to/yard
```

stage 2 submission

OMA stage 1 will produce a slurm output file for each partition in the array. Once stage 1 is done running, check the slurm output for the last of the array partitions to complete. It may include a line similar to the following:

```
** The following file(s) is (are) not yet completed:
Cache/AllAll/HUMAN/MCF7/part_1000-1712
Cache/AllAll/HUMAN/MCF7/part_1667-2855
Cache/AllAll/YEASO/MCF7/part_666-1554
** If no other process is running, delete these files and restart.
```

If it does, not every output file for stage 1 was completed. You must remove any uncompleted files and resubmit. For example, you would delete the relevant files in this manner:

```
cd path/to/StrokeofGenus/OMA/omaDir

rm Cache/AllAll/HUMAN/MCF7/part_1000-1712
rm Cache/AllAll/HUMAN/MCF7/part_1667-2855
rm Cache/AllAll/YEASO/MCF7/part_666-1554
```

Edit the `trinity_to_splice_gen_oma.sh` script, comment out the top `sbatch` command (stage 1), uncomment the second `sbatch` command (stage 2), and resubmit.

```
#sbatch $scripts/trinity_to_splice_gen_oma_stage_1.sh $output_folder $scripts $yard

sbatch $scripts/trinity_to_splice_gen_oma_stage_2.sh $output_folder $scripts $yard
```

The remaining files from OMA step 1 will be completed and then the pipeline will move on to OMA stage 2.

(If you later want to rerun stage 1, make sure the top sbatch command (stage 1) is uncommented and the bottom command (stage 2) is commented out.)

This step will produce the following outputs:

- `path/to/StrokeofGenus/OMA/omaDir/Output/OrthologousMatrix.txt`
- `path/to/StrokeofGenus/OMA/omaDir/Output/Map-SeqNum-ID.txt`

5. Step 3: identify pattern conservation

Next is to test for conservation of gene expression programs across datasets. Step 3 of StrokeofGenus consists of `trinity_to_splice_gen_ProjectR.sh`.

First, edit the header of `trinity_to_splice_gen_ProjectR.sh` to your appropriate queue and resource needs. Then, fill in the variables including pattern species, target species, and pattern number and submit.

```
# Fill in these variables to your specific use-case

pattern_species=MCF7 # species from which you take the pattern

target_species=MCF7 # species interrogated for presence of the above pattern

pattern_num=3 # number of patterns identified for the pattern species

output_folder=path/to/StrokeofGenus/

scripts=path/to/StrokeofGenus/scripts/
```

This step will produce the following outputs:

- `path/to/StrokeofGenus/MCF7/ProjectR/ProjectR_MCF7_3_into_MCF7.pdf`
- `path/to/StrokeofGenus/MCF7/ProjectR/ProjectR_MCF7_3_into_MCF7.txt`

Use the pdf file to visually assess differential conservation of gene expression patterns. Us the txt file for additional visualizations and statistical assessments of differential conservation. For in-depth examples of use-cases and output interpretation, please refer to the StrokeofGenus manuscript.

Please cite StrokeofGenus:

Identification of conserved gene expression programs activated in multiple modes of torpor across vertebrate clades. Kurt Weir, Natasha Vega, Veronica F. Busa, Ben Sajdak, Les Kallestad, Dana Merriman, Krzysztof Palczewski, Joseph Carroll, Seth Blackshaw. bioRxiv 2023.11.29.569284; doi: https://doi.org/10.1101/2023.11.29.569284

Direct any questions to kurt.weir@embl.de