1. **Introduction**

GeneTEFlow is a reproducible and platform-independent workflow, for the comprehensive analysis of gene and locus-specific TEs expression from RNA-Seq data using Nextflow and Docker technologies.

1. **Installation**

**Section 1: Install docker and singularity (*need “root” permission*)**

**Step 1:**

Installation of Docker on Ubuntu Linux system:

# apt-get install docker-ce

# docker --version

Docker version 18.03.1-ce, build 9ee9f40

# which docker

/usr/bin/docker

**Step 2:**

Installation of Singularity on Ubuntu Linux system:

# apt-get install singularity-container

# singularity --version

2.5.1-master.gd6e81547

# which singularity

/usr/local/bin/singularity

**Section 2: Getting GeneTEflow from github:**

# git clone <https://github.com/zhongw2/GeneTEFlow>

**Section 3: Build images (*need “root” permission*)**

**Using Dockerfile of GeneTEFlow.Process as an example:**

# cd GeneTEFlow\_Dockerfiles/GeneTEFlow.Process/

# docker build -t rnaseq\_pipeline.app .

Ref: <https://docs.docker.com/engine/reference/commandline/build/>

**Optional:**

If you need to run containers by Singularity, another step is required to convert docker images to Singularity images:

# cd /mnt/

# docker run -v /var/run/docker.sock:/var/run/docker.sock -v /mnt:/output --privileged -t --rm singularityware/docker2singularity rnaseq\_pipeline.app

Ref: <https://github.com/singularityware/docker2singularity>

The output file is a Singularity container under /mnt directory. For example, filename is “rnaseq\_pipeline.app-2020-3-29-cf77fe9d8630.simg”.

You may rename it, for example, to “rnaseq\_pipeline.hpc.simg” and run it on High Performance Computing (HPC) clusters by Singularity.

**Section 4: Testing containers**

Testing the docker container:

$ docker run rnaseq\_pipeline.app ls /RANSeq

Ref: <https://docs.docker.com/engine/reference/commandline/run/>

Testing the Singularity container:

$ singularity exec rnaseq\_pipeline.hpc.simg ls /RANSeq

Ref: <https://singularity.lbl.gov/docs-run>

**Section 5: install Nextflow**

**Optional:**

You might need to create a new user account for running nextflow. For instance, create a user account with name: “geneteflow1”:

# useradd -m geneteflow1 -d /mnt/geneteflow1 -s /bin/bash

# passwd geneteflow1 (geneteflow123)

**Step 1:**

Login as user geneteflow1, and install Nextflow on Ubuntu Linux system:

$ cd ~

$ pwd

/mnt/geneteflow1

$ curl -s https://get.nextflow.io | bash

$ ./nextflow run hello

Ref: <https://www.nextflow.io/>

1. **Running GeneTEFlow**

**Section 1: download reference genome and gtf files**

Human reference genome UCSC hg38 with the gene annotation (.gtf) were downloaded from illumina iGenomes collections : <https://support.illumina.com/sequencing/sequencing_software/igenome.html>

$ wget <http://igenomes.illumina.com.s3-website-us-east-1.amazonaws.com/Homo_sapiens/UCSC/hg38/Homo_sapiens_UCSC_hg38.tar.gz>

$ tar xzvf Homo\_sapiens\_UCSC\_hg38.tar.gz

$ cp Homo\_sapiens/UCSC/hg38/Sequence/WholeGenomeFasta/genome.fa .

$ cp Homo\_sapiens/UCSC/hg38/Annotation/Genes/genes.gtf .

**Section 2: collect all illumia raw data (.fastq.gz) into one folder**

$ mkdir RAW\_DATA/

You may use “ln -s” command to create the soft links to the original locations of raw data.

Here human RNA sequencing data were downloaded through GEO accession number GSE30352, including brain, heart, and testis data with biological replicates.

|  |  |  |
| --- | --- | --- |
| **Samples** | **GEO number** | **SRR number** |
| Brain replicate 1 | GSM752691 | SRR306838 |
| Brain replicate 2 | GSM752694 | SRR306841 |
| Brain replicate 3 | GSM752692 | SRR306839 |
| Heart replicate 1 | GSM752699 | SRR306847 |
| Heart replicate 2 | GSM752701 | SRR306850 |
| Testis replicate 1 | GSM752707 | SRR306857 |
| Testis replicate 2 | GSM752708 | SRR306858 |

To build small testing data sets, first 1,000,000 reads in each sample was used here.

$zcat ~/original\_locations/hsa.br.F.1\_GSM752691\_R1.fastq.gz |head -n 4000000|gzip > RAW\_DATA/hsa.br.F.1\_GSM752691\_R1.fastq.gz

**Section 3: modify the GeneTEFlow configuration file coordinately**

**Optional 1:** **configuration file for docker container**

GeneTEFlow can be run locally by specifying it in the configuration file:

process.executor = 'local'

GeneTEFlow provides functions to process both single-end and paired-end reads respectively. Please see “geneTEflow.SE.docker.config” and “geneTEflow.PE.docker.config”.

**Single-end reads:**

For example,

Specify the location of RAW data:

params.reads = "./RAW\_DATA/\*\_R1.fastq.gz"

Specify the details of samples information:

params.sampleinfoxlsx = "SE\_Nextflow\_pipeline.Human\_data.xlsx"

**Paired-end reads:**

For example,

Specify the location of RAW data:

params.reads = "./RAW\_DATA/\*\_R{1,2}.fastq.gz"

Specify the details of samples information:

params.sampleinfoxlsx = "PE\_sampledetail.xlsx"

**Optional 2:** **configuration file for Singularity container**

GeneTEFlow can be run on HPC LSF system by specifying in the configuration:

process.executor = 'lsf'

GeneTEFlow provides functions can process both single-end and paired-end reads. Please see “geneTEflow.SE.Singularity.config” and “geneTEflow.PE.Singularity.config”.

**Single-end reads:**

For example,

Specify the location of RAW data:

params.reads = "./RAW\_DATA/\*\_R1.fastq.gz"

Specify the details of samples information:

params.sampleinfoxlsx = "SE\_Nextflow\_pipeline.Human\_data.xlsx"

**Paired-end reads:**

For example,

Specify the location of RAW data:

params.reads = "./RAW\_DATA/\*\_R{1,2}.fastq.gz"

Specify the details of samples information:

params.sampleinfoxlsx = "PE\_sampledetail.xlsx"

**Please refer more details of configurations on** <https://www.nextflow.io/docs/latest/executor.html>

**Section 4: running GeneTEFlow**

**Optional 1:** **running GeneTEFlow by interacting with docker containers**

**Single-end reads:**

$ ~/nextflow run ~/GeneTEflow\_pipelines/pipeline.SE.nf -c ~/GeneTEflow\_pipelines/geneTEflow.SE.docker.config -with-dag flowchart.html -with-report nf.report.html -with-timeline nf.timeline.html

**Paired-end reads:**

$ ~/nextflow run ~/GeneTEflow\_pipelines/pipeline.PE.nf -c ~/GeneTEflow\_pipelines/geneTEflow.PE.docker.config -with-dag flowchart.html -with-report nf.report.html -with-timeline nf.timeline.html

**Optional 2:** **running GeneTEFlow by interacting with Singularity containers**

**Single-end reads:**

$ ~/nextflow run ~/GeneTEflow\_pipelines/pipeline.SE.nf -c ~/GeneTEflow\_pipelines/geneTEflow.SE.Singularity.config -with-dag flowchart.html -with-report nf.report.html -with-timeline nf.timeline.html

**Paired-end reads:**

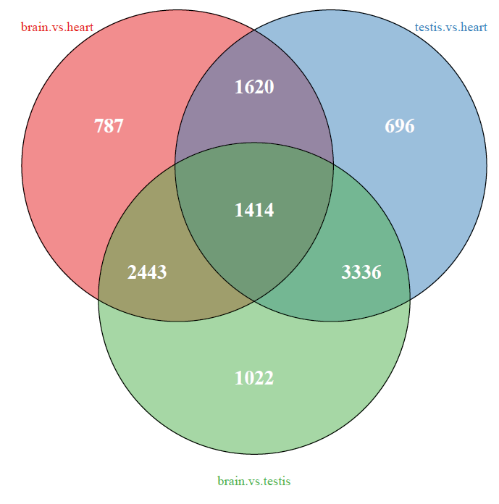
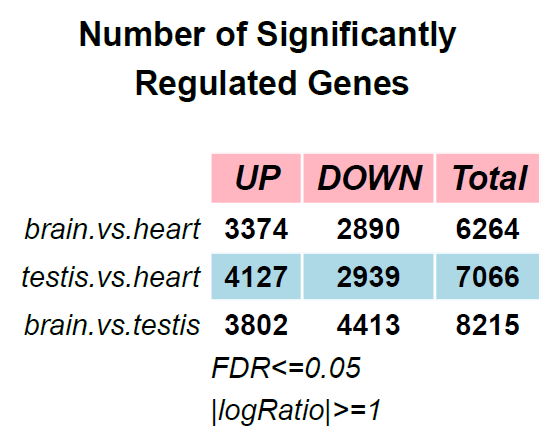
$ ~/nextflow run ~/GeneTEflow\_pipelines/pipeline.PE.nf -c ~/GeneTEflow\_pipelines/geneTEflow.PE.Singularity.config -with-dag flowchart.html -with-report nf.report.html -with-timeline nf.timeline.html

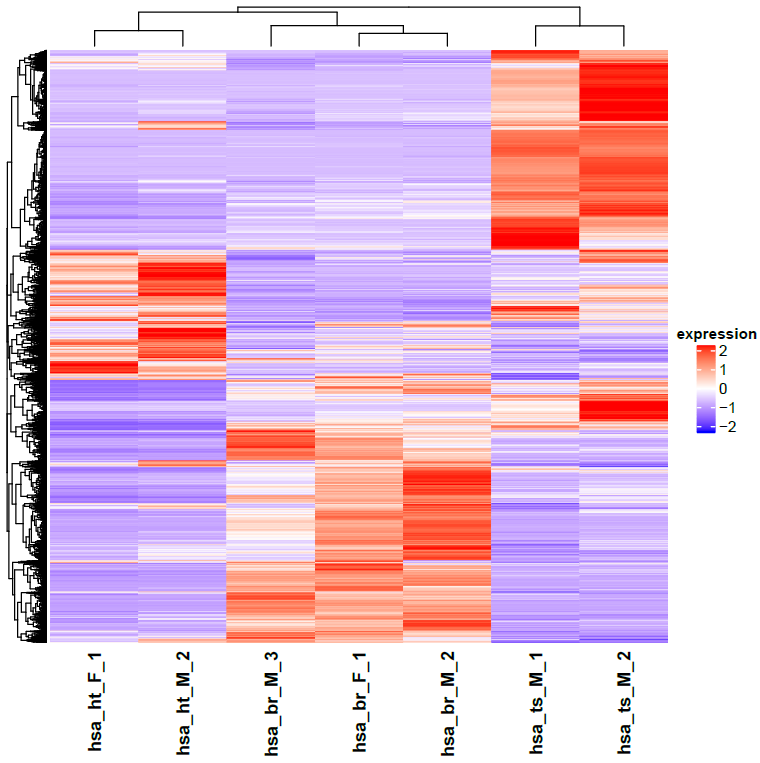
**Section 5: Results generated by GeneTEFlow**

Please see several examples in the Tests\_logfiles folder.

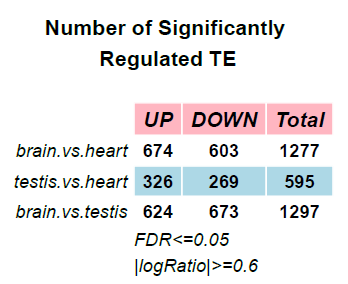
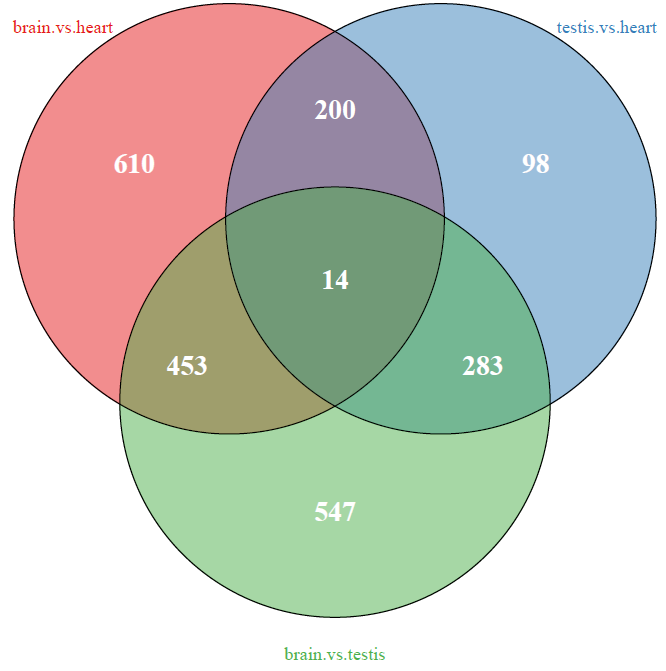
Here human RNA sequencing data (GSE30352) were used as one example.

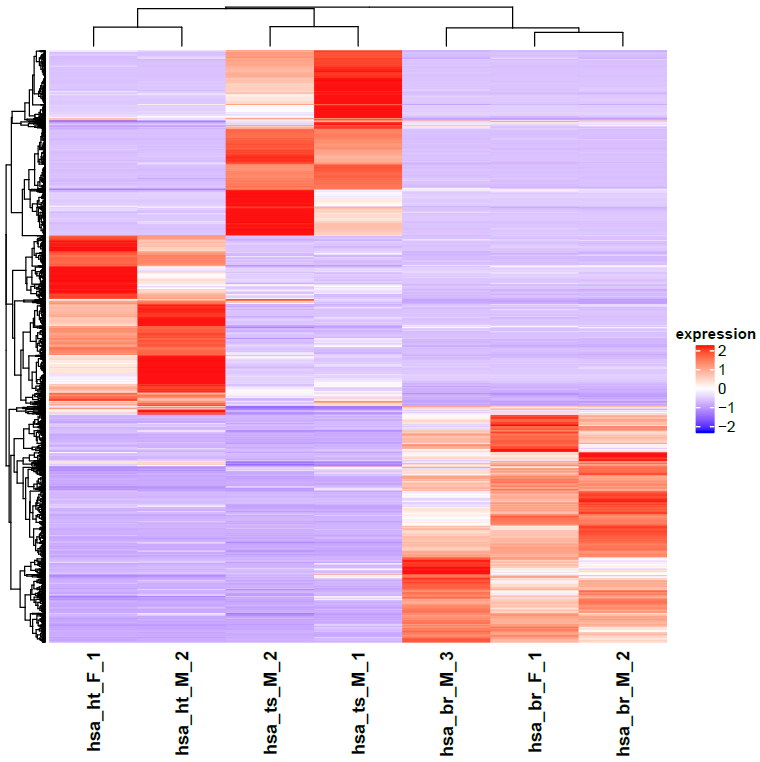
**Significantly regulated genes identified by GeneTEFlow:**



**Significantly regulated transposable elements identified by GeneTEFlow:**

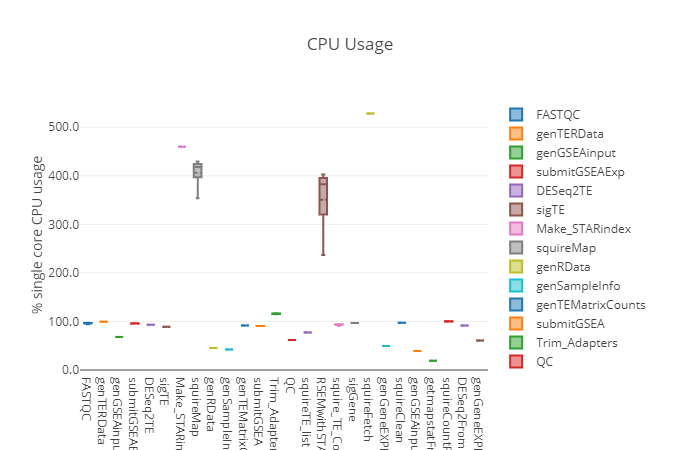
 



**Section 6: Log files generated by GeneTEFlow**

**GeneTEFlow generates three major log files: nf.report.html, nf.timeline.html, and flowchart.html.**

**One example is shown here from nf.report.html:**



Basic writing and formatting syntax of GitHub

<https://help.github.com/en/github/writing-on-github/basic-writing-and-formatting-syntax>