DEGoldS v1.0

Differential **E**xpression analysis pipelines benchmarking workflow based on **Gold** **S**tandard construction

https://github.com/GGFHF/DEGoldS

Table of contents

1. DISCLAIMER 3
2. DEPENDENCIES 4
Bash scripts (.sh files) 4
Miniconda3 4
FastQC (Bioconda installation) 4
Trinity (Bioconda installation) 5
BUSCO (Bioconda installation) 5
QUAST (Bioconda installation) 5
Bowtie2 (Bioconda installation) 5
NGShelper 5
RSEM 5
GffCompare 5
R scripts (.R files) 6
DESeq2 6
edgeR 6
Txiimport 6
readr 6
IHW 6
Apeglm 6
3. WORKFLOW PROCEDURE 7
General description 7
Gold-standard construction 7
Scripts considerations 8
4. SCRIPTS DESCRIPTION 9
5. VARIABLES INSTRUCTIONS 12
In Bash scripts 12
In R scripts 14

1. DISCLAIMER

DEGoldS is available for free download from the GitHub software repository https://github.com/GGFHF/DEGoldS) under GNU General Public License v3.0. This software has been developed in memoriam of Dr. Pablo Goikoetxea who initiated this research along with researchers from NEIKER, Departamento de Sistemas y Recursos Naturales (Universidad Politécnica de Madrid) and Universidad del País Vasco (UPV/EHU).

2. DEPENDENCIES

Bash scripts (.sh files)

The following software is necessary to run Bash scripts (.sh files). The scripts are designed to use the mentioned programs from the Bioconda repositories (using for example Miniconda3); however, any other installation should work as long as the software (and its dependencies) directories are included in the PATH variable. In this case the following lines in the scripts could be commented (#) to avoid errors and warnings:

```
#cd "$CONDA_DIR"

#source activate "$FASTQC_ENV"
```

And the following line should be added to include the directory hosting the desired program in the PATH variable:

```
PATH=/PATH_TO_YOUR_APP_DIRECTORY:$PATH
```

The instructions to install the software through Bioconda respositories using Miniconda are as follows

Miniconda3

```
$ cd /PATH_TO_YOUR_APPS_DIRECTORY (where you will install Miniconda3)

$ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh

$ chmod u+x Miniconda3-latest-Linux-x86_64.sh

$ ./Miniconda3-latest-Linux-x86_64.sh -b -p

/PATH_TO_YOUR_APPS_DIRECTORY/Miniconda3

$ rm Miniconda3-latest-Linux-x86_64.sh

$ cd /PATH_TO_YOUR_APPS_DIRECTORY/Miniconda3/bin

$ ./conda config --add channels defaults

$ ./conda config --add channels conda-forge

$ ./conda config --add channels r

$ ./conda config --add channels bioconda

$ cd /PATH_TO_YOUR_APPS_DIRECTORY/Miniconda3/bin

$ conda install --yes --name base mamba
```

FastQC (Bioconda installation)

```
$ cd /PATH_TO_YOUR_APPS_DIRECTORY/Miniconda3/bin

$ ./mamba create --yes --name fastqc fastqc
```

Trinity (Bioconda installation)

```
$ cd /PATH_TO_YOUR_APPS_DIRECTORY/Miniconda3/bin

$ ./mamba create --yes --name trinity trinity
```

BUSCO (Bioconda installation)

```
$ cd /PATH_TO_YOUR_APPS_DIRECTORY/Miniconda3/bin

$ ./mamba create --yes --name busco busco
```

QUAST (Bioconda installation)

```
$ cd /PATH_TO_YOUR_APPS_DIRECTORY/Miniconda3/bin

$ ./mamba create --yes --name quast quast
```

Bowtie2 (Bioconda installation)

```
$ cd /PATH_TO_YOUR_APPS_DIRECTORY/Miniconda3/bin

$ ./mamba create --yes --name bowtie2 bowtie2
```

NGShelper

Installation instruction in https://github.com/GGFHF/NGShelper. These instructions don´t make NGShelper available from PATH variable. However, this is not necessary as software directory has to be specified in NGSHELPER_DIR variable in the scripts.

RSEM

Although RSEM can be installed also from Bioconda repositories, issues regarding to reads simulations have been experienced. For this reason, installation from source code is suggested (instruction in https://github.com/deweylab/RSEM). These instructions make RSEM available from PATH variable, necessary to run the scripts as they are.

GffCompare

Although GffCompare can be installed also from Bioconda repositories some issues have been experienced when running. For this reason, installation from source code is suggested (instruction in https://github.com/gpertea/gffcompare). These instructions don´t make GffCompare available from PATH variable. However, this is not necessary as software directory has to be specified in GFFCOMPARE_DIR variable in the scripts.

R scripts (.R files)

The following R packages are necessary to run R scripts (.R files). All scripts have been tested in R version 4.

DESeq2

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

   install.packages("BiocManager")

BiocManager::install("DESeq2")
```

edgeR

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

   install.packages("BiocManager")

BiocManager::install("edgeR")
```

Tximport

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

   install.packages("BiocManager")

BiocManager::install("tximport")
```

readr

```
install.packages("readr")
```

IHW

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

   install.packages("BiocManager")

BiocManager::install("IHW")
```

Apeglm

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

   install.packages("BiocManager")

BiocManager::install("apeglm")
```

3. WORKFLOW PROCEDURE

General description

DEGoldS (Differential Expression analysis pipelines benchmarking workflow based on Gold Standard construction) is divided in 4 main steps which are divided at the same time in different substeps consisting on different scripts. It is suggested to create a different directory for every script during the workflow. This simulation-based workflow is designed for testing DE pipelines using two group of samples and 3 samples per group starting from real reads so results can be optimized for a particular set of samples. Although any DE analyser could be used for benchmarking this workflow pretends to use DESeq2 and edgeR, so in case user wants to test any other new scripts should be used. This workflow requires that the pipeline being tested outputs a GTF/GFF assembly (e. g. Stringtie) so transcripts IDs conversion can be done. If this is not possible a transcriptome to genome alignment with GMAP is suggested (e. g. script 1d-03) as it can output results in GFF file.

Two different simulation pipelines covering from a simpler and less realistic simulation (Sim1) to a more realistic but more complex one (Sim2) are described. Both procedures share all the steps except step2b as Sim2 gets expression values from real data while Sim1 has unrealistic expression values with no dispersion between samples. Step2b consists on 4 modules: A-selectable transcripts pool construction, B-helper scripts for selectable counts pool matrix construction, C-Sim1 TPM tables construction and D-Sim2 TPM tables construction. "A" module is used before either "C" (Sim1) or "D" (Sim2) modules, however "B" module is only used before "D" module. "B" module together with 2b-D01, 2b-D02, 2b-D03 and 2b-D04 helps to get a reliable gold-standard.

Gold-standard construction

The gold-standard is built from differential gene expression analysis (scripts 2d) of the count values taken from script 2c-01. At this point the gold-standard is called “simulation gold-standard” (simGS) because gene and transcripts IDs belong to the transcriptome where simulated reads come from. On the contrary, when this IDs are converted into IDs from the assembly of the tested DE pipeline (script 4a-02) is called “pipeline gold-standard” (pipelineGS). A reliable or suitable gold-standard for benchmarking means that simGS contains most of the “upregulated” transcripts requested in simulation but does not contain any “filler” transcripts as these should not show any significant differences in a DE expression analysis. Using real expression values (i.e. Sim2) could lead into a gold-standard with upregulated genes other than the desired ones or hiding those desired ones. Provided scripts optimize the chances (so the times to repeat the procedure) to get a count matrix with potential interesting real counts for upregulated and filler transcripts for DESeq2 and edgeR. However, after every simulation simGS should be checked to make sure that most of the genes corresponding to upregulated transcripts are recovered while no genes corresponding to filler transcripts are in it (scripts 2d). The rationale of this step consists on removing counts from a real matrix that could be in the limit between significant and non-significant differentially expressed values so increasing chances for getting a suitable gold-standard. The only thing the user has to do is to define the number of sequences (line of count in a matrix) to remove up and down significance limit according to the total number of transcripts within the transcriptome (script 2b-D02). This would reduce the times a simulation has to be performed until a suitable gold-standard is reached.

It is important to make sure that when converting simGS into pipelineGS there are no multiple pipeline genes corresponding to simulated “upregulated” transcripts in file “03-DESeq2_simGS_correspondence_table.csv” in script 4a-02. This should not be usual but in case this happens user should decide how to proceed with benchmarking process. One solution could be to consider all the possible genes but counting +1TP (True Positive) in the scoring process.

Scripts considerations

The scripts have to be only modified in the section "Variables definition" with the required strings provided. The strings should be placed between quotes (“”). Usually strings are characters strings, but if specified in instructions a number should be placed. If there are no quotes it means a number should be placed. Explanation of every variable can be seen in section 5 (variables instructions) as well as in the upper part of every script.

4. SCRIPTS DESCRIPTION

Step 1 -> Transcriptome assembly and alignment to reference genome

Script 1a-01: performs QC on input reads with FASTQC

Script 1b-01: assembles input reads into a de novo transcriptome with Trinity

Script 1b-02: filtrates de novo transcripts by lenght with NGShelper

Script 1c-01: assess transcriptome quality with QUAST

Script 1c-02: assess transcriptome completeness with BUSCO

Script 1d-01: builds reference genome index for GMAP

Script 1d-02: aligns transcriptome against reference genome with GMAP

Script 1d-03: aligns transcriptome against reference genome with GMAP (GFF output)

Script 1d-04: classifies de novo transcriptome transcripts into 0 times aligned (path 0), uniquely aligned (path 1) and multiple times aligned (path n) transcripts

Script 1d-05: modifies GFF output from GMAP for unstranded transcriptomes

Step 2 -> Simulation of reads files

Script 2a-01: performs learning step from imput reads and de novo transcriptome with RSEM

Script 2b-A01: classifies transcripts according to locus in the reference genome

Script 2b-A02: gets transcripts pools for "upregulated" and "filler" transcripts selection

Script 2b-B01: transcripts DE analysis with DESeq2 from leanrnig step data

Script 2b-B02: transcripts DE analysis with DESeq2 from leanrnig step data with no pre-filtering step

Script 2b-B03: transcripts DE analysis with edgeR from leanrnig step data

Script 2b-B04: transcripts DE analysis with edgeR from leanrnig step data with no pre-filtering step

Script 2b-B05: classifies common/not common significant transcripts in DE analysis in scripts 2b-B01 and 2b-B02.

Script 2b-C01: selects upregulated transcripts (only for Sim1)

Script 2b-C02: checks that upregulated transcripts belong to different genes (only for Sim1)

Script 2b-C03: selects filler transcripts and builds the new TPM tables for reads simulation (only for Sim1)

Script 2b-C04: performs reads simulation with RSEM (only for Sim1)

Script 2b-D01: gets count matrix building from 2a-01 output (only for Sim2)

Script 2b-D02: builds the selectable count pool matrix (only for Sim2)

Script 2b-D03: selects significant and not significant counts (only for Sim2)

Script 2b-D04: builds the significant and not significant count matrix (only for Sim2)

Script 2b-D05: selects upregulated transcripts (only for Sim2)

Script 2b-D06: checks that upregulated transcripts belong to different genes (only for Sim2)

Script 2b-D07: selects filler transcripts (only for Sim2)

Script 2b-D08: checks that filler transcripts belong to different genes (only for Sim2)

Script 2b-D09: obtains transcripts from transcriptome that will have 0 counts -surplus transcripts- (only for Sim2)

Script 2b-D10: prepare counts tables (isoforms.results files) for 2b-D11 (only for Sim2)

Script 2b-D11: builds the new TPM tables for reads simulation (only for Sim2)

Script 2b-D12: performs reads simulation with RSEM (only for Sim2)

Script 2c-01: generates simulated reads from de novo transcriptome with RSEM

Script 2d-01: performs DE analysis from simulation stats to build simulation gold-standard with DESeq2 (simGS)

Script 2d-02: performs DE analysis from simulation stats to build simulation gold-standard with edgeR (simGS)

Step 3 -> To be tested pipeline running

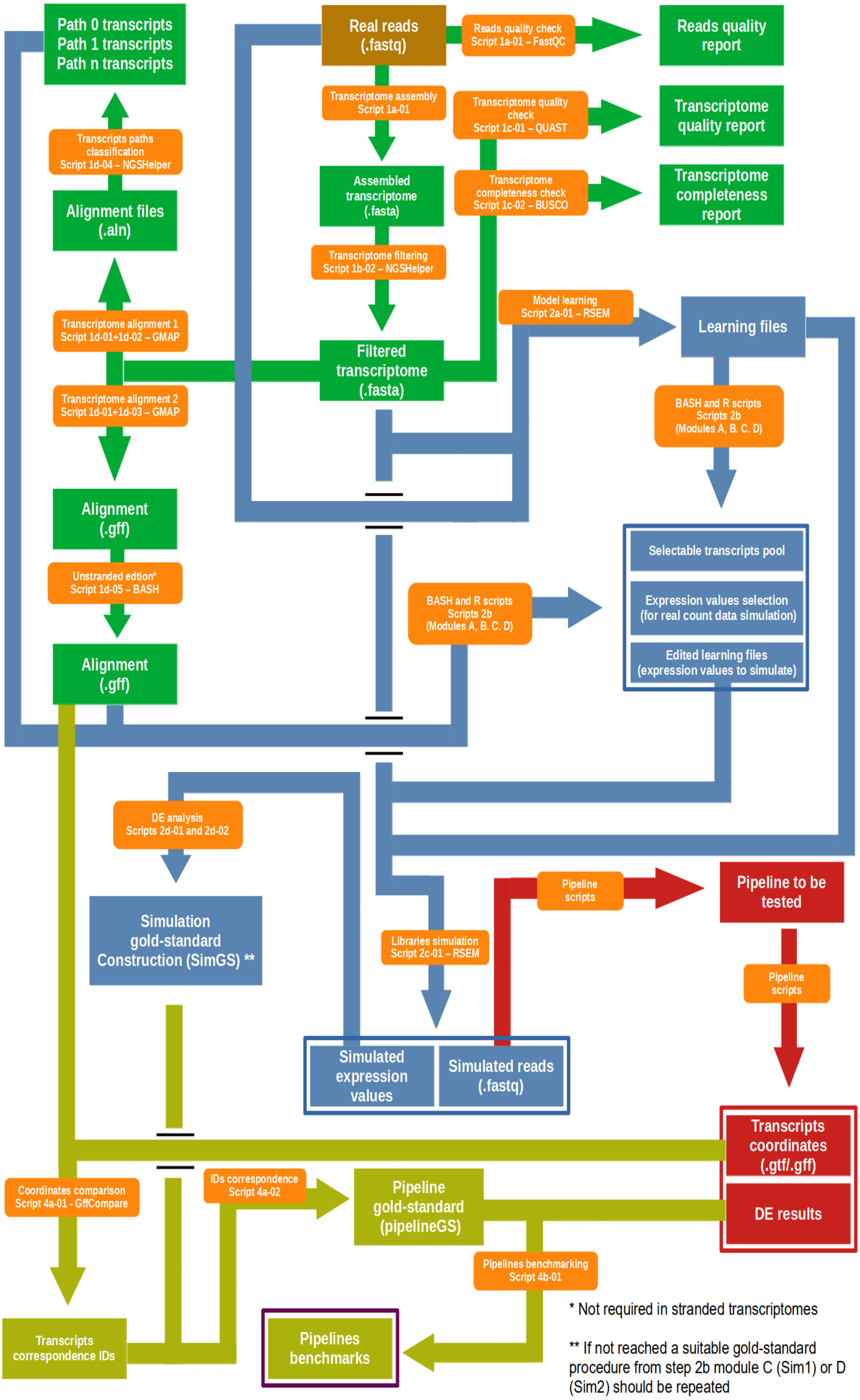
In this step each user should include its own scripts from the DE pipeline to compare.

Step 4 -> Benchmarking of the tested DE analysis pipeline

Script 4a-01: gets coordinates correspondence between de novo transcripts and transcripts assembled in tested pipeline using GFF output from GMAP alignment and GTF assembly annotation using GffCompare

Script 4a-02: outputs pipeline gold-standard (pipelineGS) from the correspondance in script 4a-01 and simGS in step 2d

Script 4b-01: performs benchmarking of the pipeline according to pipelineGS and results in DE analysis in the tested pipeline


Path 0 transcripts
Path 1 transcripts
Path n transcripts
Real reads (.fastq)
Reads quality check Script 1a-01 – FastQC
Reads quality report
Transcripts paths classification Script 1d-04 – NGSHelper
Transcriptome assembly Script 1a-01
Transcriptome quality check Script 1c-01 – QUAST
Transcriptome quality report
Alignment files (.aln)
Assembled transcriptome (.fasta)
Transcriptome completeness check Script 1c-02 – BUSCO
Transcriptome completeness report
Transcriptome filtering Script 1b-02 – NGSHelper
Model learning Script 2a-01 – RSEM
Learning files
Transcriptome alignment 1 Script 1d-01+1d-02 – GMAP
Filtered transcriptome (.fasta)
Transcriptome alignment 2 Script 1d-01+1d-03 – GMAP
BASH and R scripts Scripts 2b (Modules A, B. C. D)
Alignment (.gff)
Selectable transcripts pool
BASH and R scripts Scripts 2b (Modules A, B. C. D)
Expression values selection (for real count data simulation)
Unstranded edtion* Script 1d-05 – BASH
Edited learning files (expression values to simulate)
Alignment (.gff)
DE analysis Scripts 2d-01 and 2d-02
Pipeline scripts
Pipeline to be tested
Simulation gold-standard Construction (SimGS) **
Libraries simulation Script 2c-01 – RSEM
Pipeline scripts
Simulated expression values
Simulated reads (.fastq)
Transcripts coordinates (.gtf/.gff)
Coordinates comparison Script 4a-01 - GffCompare
IDs correspondence Script 4a-02
Pipeline gold-standard (pipelineGS)
DE results
Pipelines benchmarking Script 4b-01
Transcripts correspondence IDs
Pipelines benchmarks

* Not required in stranded transcriptomes

** If not reached a suitable gold-standard procedure from step 2b module C (Sim1) or D (Sim2) should be repeated

5. VARIABLES INSTRUCTIONS

In Bash scripts

BOWTIE2_DIR: Bowtie2 directory

BUSCO_ENV: BUSCO environment in Conda

COMMONDEIDS: "03-DESeq2_edgeR_common_sig_isoforms.csv" file path from script 2b-B05

CONDA_DIR: Conda directory

CORRESP_FILE_PATH: "ref_pipeline.txt" file path from 4a-01

COUNTMATRIX: "01-learn_count_matrix.csv" file path from script 2b-D01

DE_RES_FILE_DEL: column delimiter in $DE_RES_FILE_PATH

DE_RES_FILE_GENE_ID_COL_NUMBER: gene ID column number in $DE_RES_FILE_PATH

DE_RES_FILE_PATH: DE analysis results table from tested pipeline

DESEQ2_LEARN_DE_ALL_RES: "01-learning_DESeq2_all_seqs.csv" file path from script 2b-B01

DESEQ2_LEARN_DE_SIG_RES: "02-learning_DESeq2_sig_seqs.csv" file path from script 2b-B01

EDGER_LEARN_DE_ALL_RES: "01-learning_edgeR_all_seqs.csv" file path from script 2b-B01

EDGER_LEARN_DE_SIG_RES: "02-learning_edgeR_sig_seqs.csv" file path from script 2b-B03

EDITED_GMAP_GFF_FILE: edited GMAP GFF output (unstranded_edited_alignment.gff in 1d-05)

EDITED_SAMPLES_LEARN_DIR: directory with generated "isoforms.results" in script 2b-C03 (Sim1) or 2b-D11 (Sim2)

FASTQC_ENV: FASTQC environment in Conda

FILLER_SEL_SEQS_FILE_PATH: "filler_seqs.txt" file path from script 2b-D07

FILTNUMNOSIGDESEQ2: number of DESeq2 non-significant sequences to be filtered starting from head

FILTNUMNOSIGEDGER: number of edgeR non-significant sequences to be filtered starting from head

FILTNUMSIGDESEQ2: number of DESeq2 significant sequences to be filtered starting from bottom

FILTNUMSIGEDGER: number of edgeR significant sequences to be filtered starting from bottom

GENOME_FILE_PATH: gziped reference genome FASTA file path

GFFCOMPARE_DIR: GffCompare directory

GMAP_ALIGNMENT_FILE: (alignment.log in 1d-02)

GMAP_ENV: GMAP environment in Conda

GMAP_GFF_FILE: GMAP GFF output (alignment.gff in 1d-03)

GMAP_INDEX_DIR: reference genome GMAP index directory

GMAP_INDEX_NAME: reference genome GMAP index name

GROUP01_UPREG_SEQS_X1_FILE_PATH: "group01_upreg_seqs_x1.txt" file path from 2b-C01

GROUP01_UPREG_SEQS_X2_FILE_PATH: "group01_upreg_seqs_x2.txt" file path from 2b-C01

GROUP01_UPREG_SEQS_X3_FILE_PATH: "group01_upreg_seqs_x3.txt" file path from 2b-C01

GROUP01_UPREG_SEQS_X4_FILE_PATH: "group01_upreg_seqs_x5.txt" file path from 2b-C01

GROUP01_UPREG_SEQS_X5_FILE_PATH: "group01_upreg_seqs_x4.txt" file path from 2b-C01

GROUP02_UPREG_SEQS_X1_FILE_PATH: "group01_upreg_seqs_x1.txt" file path from 2b-C01

GROUP02_UPREG_SEQS_X2_FILE_PATH: "group02_upreg_seqs_x2.txt" file path from 2b-C01

GROUP02_UPREG_SEQS_X3_FILE_PATH: "group03_upreg_seqs_x3.txt" file path from 2b-C01

GROUP02_UPREG_SEQS_X4_FILE_PATH: "group04_upreg_seqs_x4.txt" file path from 2b-C01

GROUP02_UPREG_SEQS_X5_FILE_PATH: "group05_upreg_seqs_x5.txt" file path from 2b-C01

GS_NAME: gold-standard given name (recommended a name distinguishing each DE analysers, i.e. DESeq2/edgeR)

LEARN_DIR: "02-model_learning/output/" directory from script 2a-01

LEFT_SUFFIX: left reads file suffix (including extension)

LOCI_FILE_PATH: "trancriptome.loci" file from script 2b-A01

MEMORY: maximum number of memory to use (in GB)

NCPU: number of CPUs

NF_DESEQ2_LEARN_DE_SIG_RES: "02-nf_learning_DESeq2_sig_seqs.csv" file path from script 2b-B01

NF_EDGER_LEARN_DE_SIG_RES: "02-learning_edgeR_sig_seqs.csv" file path from script 2b-B01

NGSHELPER_DIR: NGSHelper directory

NO_SIG_COUNT_MATRIX_FILE_PATH: "06-no_sig_rest_removed_count_matrix.csv" file path from script 2b-D02

NOTCOMMONDEIDS: "06-DESeq2_edgeR_not_common_sig_isoforms.csv" file path from script 2b-B05

PATH1_SEQS_IDS_FILE_PATH: "assembly-ids-1path.txt" from script 1d-04

PIPELINE_ANN_FILE: tested pipeline GTF/GFF file

PIPELINE_TX2GENE_FILE_PATH: two column text file (transcript ID-gene ID) separated by space (view tx2gene.csv example file) for pipeline gene IDs

PIPELINEGS_FILE_PATH: "05-$GS_NAME_pipelineGS_geneIDs.txt" file path from script 4a-02

QUAST_ENV: QUAST environment in Conda

RAW_TRANSCRIPTOME_FILE_PATH: pre-filtered transcriptome FASTA file path

READS_DIR: Reads directory

READSNUMBER: number of reads to be simulated

REPNUMBER: number of replicates to be simulated for each input "isoforms.results" file

RIGHT_SUFFIX: right reads file suffix (including extension)

SAMPLE_LEARN_FILE_PATH: one of the ".isoform.result" files file path in 2a-01

SEL_NO_SIG_SEQS_FILE_PATH: "02-sel_no_sig_seqs.txt" file path from script 2b-D03

SEL_SIG_SEQS_FILE_PATH: "01-sel_sig_seqs.txt" file path from script 2b-D03

SIG_COUNT_MATRIX_FILE_PATH: "05-common_sig_rest_removed_count_matrix.csv" file path from script 2b-D02

SIMGS_FILE_DEL: column delimiter in $SIMGS_FILE_PATH

SIMGS_FILE_GENE_ID_COL_NUMBER: gene ID column number in $SIMGS_FILE_PATH

SIMGS_FILE_PATH: "simGS_DESeq2.csv" (DESeq2) and "simGS_edgeR.csv" file path (edgeR) from 2d-01 and 2d-02 respectively or any other used

SURPLUS_OBT_SEQS_FILE_PATH: "surplus_seqs_ids.txt" file path from script 2b-D09

TRANSCRIPTOME_FILE_PATH: by length filtered transcriptome FASTA file path

TRANSCRIPTOME_INDEX_DIR: transcriptome RSEM index directory

TRANSCRIPTOME_INDEX_NAME: transcriptome RSEM index name

TRANSCRIPTOME_NAME: filtered transcriptome name (file name without extension)

TRINITY_ENV: Trinity environment in Conda

UPREG_SEL_SEQS_FILE_PATH: "01-DE_trinity_transcripts.txt" file path from 2b-C02 (Sim1) and "upreg_seqs.txt" file path from script 2b-D05 (Sim2)

WD: Working directory

In R scripts

base_value: base expression level value

candidate_upreg_seqs_file_path: a text file with a list of selectable transcript IDs in the first column with no header for upregulated transcripts selection (i. e. selectable transcripts pool), for example, "08-path1_candidate_unique_seqs_IDs.txt" (Sim1) and "11-path1_candidate_unique-plus_seqs.txt" (Sim2) file path from 2b-A02

counts_dir: a directory where .sim.isoforms.results from script 2c-01 are placed after changing 5th column name into "expected_count" instead of "count"

filler_group01_sample01_learn_file_path: filler ".isoform.result" file path in 2b-D10 for sample 1 in group 1 filler_group01_sample02_learn_file_path: filler ".isoform.result" file path in 2b-D10 for sample 2 in group 1

filler_group01_sample03_learn_file_path: filler ".isoform.result" file path in 2b-D10 for sample 3 in group 1

filler_group02_sample01_learn_file_path: filler ".isoform.result" file path in 2b-D10 for sample 1 in group 2

filler_group02_sample02_learn_file_path: filler ".isoform.result" file path in 2b-D10 for sample 2 in group 2

filler_group02_sample03_learn_file_path: filler ".isoform.result" file path in 2b-D10 for sample 3 in group 2

filler_pre_candidate_seqs_file_path: a text file with a list of selectable transcript IDs in the first column with no header for filler transcripts selection (i. e. selectable transcripts pool), for example, "10-candidate_unique-plus_seqs.txt" file path in script 2b-A02

filler_sel_seqs_number: number of filler transcripts to select

group01_sample01_learn_file_path: ".isoform.result" file in 2a-01 for sample 1 in group 1

group01_sample01_samplename: sample name for sample 1 in group 1 (should coincide with the names of the input reads in the workflow)

group01_sample02_learn_file_path: ".isoform.result" file in 2a-01 for sample 2 in group 1

group01_sample02_samplename: sample name for sample 2 in group 1 (should coincide with the names of the input reads in the workflow)

group01_sample03_learn_file_path: ".isoform.result" file in 2a-01 for sample 3 in group 1

group01_sample03_samplename: sample name for sample 3 in group 1 (should coincide with the names of the input reads in the workflow)

group01_upreg_seqs_x1_number: number of transcripts to select for group 1 with level 1 of expression in upregulated transcripts

group01_upreg_seqs_x2_number: number of transcripts to select for group 1 with level 2 of expression in upregulated transcripts

group01_upreg_seqs_x3_number: number of transcripts to select for group 1 with level 3 of expression in upregulated transcripts

group01_upreg_seqs_x4_number: number of transcripts to select for group 1 with level 4 of expression in upregulated transcripts

group01_upreg_seqs_x5_number: number of transcripts to select for group 1 with level 5 of expression in upregulated transcripts

group01_upreg_X1_sel_seqs_file_path: selected transcripts in group 1 with level 1 of expression in upregulated transcripts ("group01_upreg_seqs_x1.txt" file path from script 2b-C01)

group01_upreg_X2_sel_seqs_file_path: selected transcripts in group 1 with level 2 of expression in upregulated transcripts ("group01_upreg_seqs_x2.txt" file path from script 2b-C01) group01_upreg_X3_sel_seqs_file_path: selected transcripts in group 1 with level 3 of expression in upregulated transcripts ("group01_upreg_seqs_x3.txt" file path from script 2b-C01)

group01_upreg_X4_sel_seqs_file_path: selected transcripts in group 1 with level 4 of expression in upregulated transcripts ("group01_upreg_seqs_x4.txt" file path from script 2b-C01)

group01_upreg_X5_sel_seqs_file_path: selected transcripts in group 1 with level 5 of expression in upregulated transcripts ("group01_upreg_seqs_x5.txt" file path from script 2b-C01)

group02_sample01_learn_file_path: ".isoform.result" file in 2a-01 for sample 1 in group 2

group02_sample01_samplename: sample name for sample 1 in group 1 (should coincide with the names of the input reads in the workflow)

group02_sample02_learn_file_path: ".isoform.result" file in 2a-01 for sample 2 in group 2

group02_sample02_samplename: sample name for sample 2 in group 2 (should coincide with the names of the input reads in the workflow)

group02_sample03_learn_file_path: ".isoform.result" file in 2a-01 for sample 3 in group 2

group02_sample03_samplename: sample name for sample 3 in group 3 (should coincide with the names of the input reads in the workflow)

group02_upreg_seqs_x1_number: number of transcripts to select for group 2 with level 1 of expression in upregulated transcripts

group02_upreg_seqs_x2_number: number of transcripts to select for group 2 with level 1 of expression in upregulated transcripts

group02_upreg_seqs_x3_number: number of transcripts to select for group 2 with level 1 of expression in upregulated transcripts

group02_upreg_seqs_x4_number: number of transcripts to select for group 2 with level 1 of expression in upregulated transcripts

group02_upreg_seqs_x5_number: number of transcripts to select for group 2 with level 1 of expression in upregulated transcripts

group02_upreg_X1_sel_seqs_file_path: selected transcripts in group 2 with level 1 of expression in upregulated transcripts ("group02_upreg_seqs_x1.txt" file path from script 2b-C01)

group02_upreg_X2_sel_seqs_file_path: selected transcripts in group 2 with level 2 of expression in upregulated transcripts ("group02_upreg_seqs_x2.txt" file path from script 2b-C01)

group02_upreg_X3_sel_seqs_file_path: selected transcripts in group 2 with level 3 of expression in upregulated transcripts ("group02_upreg_seqs_x3.txt" file path from script 2b-C01)

group02_upreg_X4_sel_seqs_file_path: selected transcripts in group 2 with level 4 of expression in upregulated transcripts ("group02_upreg_seqs_x4.txt" file path from script 2b-C01)

group02_upreg_X5_sel_seqs_file_path: selected transcripts in group 2 with level 5 of expression in upregulated transcripts ("group02_upreg_seqs_x5.txt" file path from script 2b-C01)

learn_dir: "02-model_learning/output/" directory from script 2a-01 no_sig_count_matrix_file_path: "06-no_sig_rest_removed_count_matrix.csv" file path from script 2b-D02

sample_learn_file_path: one of the ".isoform.result" files file path in 2a-01

samples_description_file_path: samples description text file path (view samples.txt example file)

sel_no_sig_count_matrix_file_path: "02-complete_no_sig_count_matrix.csv" file path from script 2b-D04

sel_no_sig_seqs_number: number of sequences to select from count matrix in "06-no_sig_rest_removed_count_matrix.csv" from script 2b-D02 that will belong to filler transcripts

sel_sig_count_matrix_file_path: "01-complete_sig_count_matrix" file path from script 2b-D04

sel_sig_seqs_number: number of sequences to select from count matrix in "05-common_sig_rest_removed_count_matrix.csv" from script 2b-D02 that will belong to upregulated transcripts

sig_count_matrix_file_path: "05-common_sig_rest_removed_count_matrix.csv" file path from script 2b-D02

sim_tx2gene_file_path: two column text file (transcript ID-gene ID) separated by space (view tx2gene.csv example file) for transcriptome gene IDs

surplus_group01_sample01_learn_file_path: surplus ".isoform.result" file path in 2b-D10 for sample 1 in group 1

surplus_group01_sample02_learn_file_path: surplus ".isoform.result" file path in 2b-D10 for sample 2 in group 1

surplus_group01_sample03_learn_file_path: surplus ".isoform.result" file path in 2b-D10 for sample 3 in group 1

surplus_group02_sample01_learn_file_path: surplus ".isoform.result" file path in 2b-D10 for sample 1 in group 2

surplus_group02_sample02_learn_file_path: surplus ".isoform.result" file path in 2b-D10 for sample 2 in group 2

surplus_group02_sample03_learn_file_path: surplus ".isoform.result" file path in 2b-D10 for sample 3 in group 2

upreg_group01_sample01_learn_file_path: upregulated ".isoform.result" file path in 2b-D10 for sample 1 in group 1

upreg_group01_sample02_learn_file_path: upregulated ".isoform.result" file path in 2b-D10 for sample 2 in group 1

upreg_group01_sample03_learn_file_path: upregulated ".isoform.result" file path in 2b-D10 for sample 3 in group 1

upreg_group02_sample01_learn_file_path: upregulated ".isoform.result" file path in 2b-D10 for sample 1 in group 2 upreg_group02_sample02_learn_file_path: upregulated ".isoform.result" file path in 2b-D10 for sample 2 in group 2

upreg_group02_sample03_learn_file_path: upregulated ".isoform.result" file path in 2b-D10 for sample 3 in group 2

upreg_sel_seqs_file_path: "01-DE_trinity_transcripts.txt" file path from script 2b-C02 (Sim1) or "upreg_seqs.txt" file path from script 2b-D05 (Sim2)

upreg_seqs_number: number of upregulated transcripts to select

upreg_value_X1: upregulated expression value for level 1

upreg_value_X2: upregulated expression value for level 2

upreg_value_X3: upregulated expression value for level 3

upreg_value_X4: upregulated expression value for level 4

upreg_value_X5: upregulated expression value for level 5

wd: Working directory