User Manual

EvoSeq\_RNA: an in-house bioinformatic framework for bulk RNA-seq analysis

Release v0.0.1

(2024-05-20)

Correspondence Contact:

Jia-Xing Yue (岳家兴)

Email: yuejiaxing[AT]gmail[DOT]com

GitHub: yjx1217

Twitter: iamphioxus, evomicslab

Website: <http://evomicslab.org/>

CONTENT

[Introduction 3](#_Toc167114979)

[License 3](#_Toc167114980)

[Software Installation and Configuration 4](#_Toc167114981)

[Installation 4](#_Toc167114982)

[The Testing Example Walking Through 5](#_Toc167114983)

[**The EvoSeq Installation** 5](#_Toc167114984)

[**Running analysis with EvoSeq** 6](#_Toc167114985)

**Introduction**

EvoSeq is an integrated computational framework for RNA-Seq data analysis. It covers all major RNA-Seq application scopes, including reads mapping, reads quantifying, and differentially expressed genes (DEGs) analysis.

Under the hood, a series of task-specific modules are provided to carry out the full workflow of EvoSeq:

* **00. RNAseq\_Reads**

quality control for the FASTQ reads

* **01. Reference\_Preprocessing**

downloading and preprocessing the reference genome and annotation

* **02. Expression\_Quantification**

mapping and quantifying reads

* **03. Expression\_Exploration**

exploring the correlation among samples & the distribution of the gene expressions

* **04. Expression\_Differentiation**

comparing gene expressions between different groups or samples

**License**

EvoSeq itself is distributed under the MIT license but some of its dependencies might have stricter licenses for commercial use. Please check the licensing details of those dependencies.

**Software Installation and Configuration**

EvoSeq relies on a number of third-party bioinformatics tools for data analysis, all of which can be automatically installed and configured by EvoSeq. A bash script (“install\_dependencies.sh”) is pre-shipped with EvoSeq to perform such installation and configuration.

**Installation**

EvoSeq is implemented in Bash, Perl, and R. It is designed for a desktop or computing server running an x86-64-bit Linux operating system. Multithreaded processors are preferred to speed up the process since many steps can be configured to use multiple threads in parallel. A stable internet connection is required for its installation.

git clone https://github.com/yjx1217/EvoSeq\_RNA.git  
cd EvoSeq\_RNA  
bash ./install\_dependencies.sh

If the installation succeeds, you should see the following message:

“EvoSeq\_RNA message: This bash script has been successfully processed! :)”

This signifies the success of the installation process.

Upon the success of the installation, a subdirectory named build and a file named env.sh will be generated. The build subdirectory holds all the installed dependencies, while the env.sh file contains the execution paths of these dependencies. This file will be automatically loaded to set up the working environment for EvoSeq’s various modules. The base directory of EvoSeq is defined as $EVOSEQ\_HOME in this file.

If an unexpected error occurs during installation, normally you can just re-do the “bash ./install\_dependencies.sh” step and the installation should be able to automatically resume from the previous interruption point.

**The Testing Example Walking Through**

**The EvoSeq Installation**

1. **Downloading and installing EvoSeq**

Run this step by typing:

git clone https://github.com/yjx1217/EvoSeq\_RNA.git  
cd EvoSeq\_RNA  
bash ./install\_dependencies.sh

**[Important Note]**

Please note that it will take a while for the installation to finish. Therefore, it is recommended to run the bash script above with nohup, which prevents the unintended interruption of the running script. The same trick applies to all the other module-specific bash script as well.

nohup bash ./install\_dependencies.sh > run.log.txt 2>&1 &

Please note if the installation script prompts for the following message at the end of the installation process:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Your java version is not the version required by EvoSeq (java v1.8)!

Please manually set the directory path to java 1.8 executable on the last line of the env.sh file generated by this installation script!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

If this message is prompted, please manually modify the last line of the env.sh file to provide the path to the java 1.8 executable accordingly after the installation process successfully finishes.

If the installation succeeds, you should see the following message:

“EvoSeq\_RNA message: This bash script has been successfully processed! :)”

This signifies the success of the installation process. The same is true for all module-specific bash scripts (named as “EvoSeq.\*.sh”) of EvoSeq.

Upon the success of the installation, a subdirectory named build and a file named env.sh will be generated. The build subdirectory holds all the installed dependencies, while the env.sh file contains the execution paths of these dependencies. This file will be automatically loaded to set up the working environment for EvoSeq’s various modules. The base directory of EvoSeq is defined as $EVOSEQ\_HOME in this file.

In case of installation failure (most likely due to an internet connection problem that might occur temporarily), the users only need to re-run the installation script install\_dependencies.sh. EvoSeq will automatically detect the previous interruption point and resume the installation process.

**Major outputs when running this step:**

build

# The subdirectory holding all the installed dependencies.

env.sh

# The file containing the execution paths of these dependencies.

**Running analysis with EvoSeq**

1. **Creating an EvoSeq project directory**

Copying the Project\_Template directory to create your own EvoSeq project directory. Here we will name it as Project\_Example for this testing example. Once created, enter into this directory.

Run this step by typing:

cp -r Project\_Template Project\_Example  
cd Project\_Example

1. **Setting up the raw FASTQ reads & checking their qualities (Module 00)**

At this step, we are going to set up the raw next-generation FASTQ reads for quality control. For this testing example, we will download the reads generated in BioProject **PRJNA928788**.

For downloading the testing FASTQ reads by typing:

cd 00.RNAseq\_Reads

bash download\_reads\_for\_testing\_example.sh

Once the FASTQ reads have been downloaded successfully, their qualities can be checked by typing:

bash EvoSeq.00.RNAseq\_Reads\_QC.sh

**Major outputs when running this step for the testing example:**

MultiQC\_outputs/multiqc\_report.html

# An HTML report integrated quality results of all samples.

1. **Setting up the reference genome (Module 01)**

For the testing example, we are going to use the human reference genome (version: GRCh38). At this step, the human reference genome assembly and annotation will be automatically downloaded from Ensembl ([https://www.ensembl.org](https://www.ensembl.org/)) and be properly set up.

EvoSeq supports all organisms with reference genome and annotation retrievable via Ensembl or its sister sites (e.g., Ensembl Fungi, Ensembl Plants, Ensembl Protists, and Ensembl Metazoa). This provided bash script is a general template of downloading and setting up the reference genome for the human or mouse. You can adapt it for your project by specifying the assembly and annotation downloading URLs of the organisms.

In EvoSeq, there is a dedicated section for customized parameter settings at the beginning of each module-specific bash script. In general, you only need to modify this part to adapt the script for your project.

Run this step by typing:

cd 01.Reference\_Preprocessing

bash EvoSeq.01.Reference\_Preprocessing.sh

**Major outputs when running this step for the testing example:**

ref.genome.fa

# The preprocessed reference genome file in FASTA format.

ref.genome.gtf

# The preprocessed annotation file in GTF format.

ref.transcriptome.fa

# The preprocessed reference transcriptome file in FASTA format.

ref.transcript2gene\_map.txt

# The preprocessed ID mapping file among transcript IDs, gene IDs, and gene names.

1. **Quantifying the reads based on the preprocessed reference genome (Module 02)**

At this step, EvoSeq will quantify the reads of each sample.

EvoSeq uses a space- or tab-delimited master sample table file (e.g., Master\_Sample\_Table.Batch\_PRJNA928788.txt for the testing example, in which “Batch\_PRJNA928788” is the specified batch\_id) to control all downstream analysis in a batch-by-batch fashion. Multiple samples from one or two comparison groups can be specified in a single master sample table file. Samples from different batches can be processed at the same time without interference. Such master sample table should contain 6 columns: “sample\_id”, “R1\_read\_file,R2\_read\_file”, “biological\_sample”, “treatment\_condition”, “sampling\_timepoint”, “biological\_replicate\_id”, and “technical\_replicate\_id”. All lines starting with “#” will be automatically ignored.

Run this step by typing:

cd 02.Expression\_Quantification

bash EvoSeq.02.Expression\_Quantification.sh

**Major outputs when running this step for the testing example:**

Batch\_PRJNA928788/Batch\_PRJNA928788.transcripts\_quant/quant.sf

# A quantified table contained estimated transcript expressions.

1. **Exploring the gene expressions among samples (Module 03)**

At this step, EvoSeq will first convert the counts from transcriptional level to gene level; and will then process the raw counts to normalized / transformed counts to compare similarities among samples.

Please edit the contents of the master table & parameters in the bash script to adapt your project first, and then run this step by typing:

cd 03.Expression\_Exploration

bash EvoSeq.03.Expression\_Exploration.sh

Once the step was performed successfully,

**Major outputs when running this step for the testing example:**

Batch\_PRJNA928788/based\_on\_normalized\_counts

# The folder contained a table showing normalized counts in each sample and plots showing similarities between samples

Completely outputs generated by the example data at this step can be previewed at EvoSeq\_RNA/Example\_Outputs/03.Expression\_Exploration/.

1. **Performing differential expression between different comparison groups (Module 04)**

At this step, EvoSeq will perform differential expression comparison between different comparison groups.

Please edit the contents of the master table and parameters in the bash script to adapt your project first, and then run this step by typing:

cd 04.Expression\_Differentiation

bash EvoSeq.04.Expression\_Differentiation.sh

**Major outputs when running this step for the testing example:**

Batch\_PRJNA928788/contrast\_DEG\_out/Batch\_PRJNA928788.contrast\_DEG.full\_table.txt

# The full differential expression table contained all genes (except for low count genes).

Batch\_PRJNA928788/contrast\_DEG\_out/Batch\_ PRJNA928788.contrast\_DEG.significant\_table.txt

# The differential expression table contained significantly expressed genes only.

Batch\_PRJNA928788/contrast\_DEG\_out/Batch\_AsPC.contrast\_DEG.volcanoplot.pdf

# The volcano plot.

Completely outputs generated by the example data at this step can be previewed at EvoSeq\_RNA/Example\_Outputs/04.Expression\_Differentiation/.