EPITENSOR MANUAL

Version 0.9

Yun Zhu

May 4, 2015

1. Overview

EpiTensor is a program that can construct 3-D interaction maps in the genome from 1-D epigenomes. It takes aligned reads in bam format as input (if one has bed-format files, he/she can easily use bedtools to convert them to bam files) and outputs promoter-promoter, promoter-enhancer, and enhancer-enhancer interaction pairs in one or more cell types.

The first step is to compute the coverage density profile for a bam file because the subsequent analysis (i.e. `epitensor.bash`) only takes coverage density profiles as input, rather than raw bam files. To compute coverage density profile, one can use the "`preprocessing.bash`" function (see demo1 in "Getting Started"). Although "`preprocessing.bash`" handles one bam file at a time, one can easily parallelize the process by submitting multiple jobs to a cluster, each involving one "`preprocessing`" function.

Once coverage density profiles are generated, one can proceed to the second step – `epitensor` (see demo2 in "Getting Started"). The "`epitensor.bash`" function takes as input a set of coverage density profiles (in rdata format) for multiple assays in multiple cell types, and performs tensor analysis to identify promoter-promoter, promoter-enhancer, and enhancer-enhancer pairs.

The third step is to extract cell-type-active interaction pairs from the promoter-promoter, promoter-enhancer, and enhancer-enhancer interaction pairs identified in the second step. One can run "`extract_active_tss_active_tss_pairs.bash`", "`extract_active_tss_active_enh_pairs.bash`", and "`extract_active_enh_active_enh_pairs.bash`" for this purpose (see demo3 in "Getting Started").

The first preprocessing step takes 5-10 mins to process one bam file on one CPU. If one has a cluster, he/she can easily parallelize the process as described above. The second step takes around 30-90 mins, depending on the size of the chromosome. The third stop usually takes less than one minute.

2. Installation

Requirements:

1. MATLAB Compiler Runtime (MCR, http://www.mathworks.com/supportfiles/MCR_Runtime/R2013a/MCR_R2013a_glnxa64_installer.zip)

2. R enviroment (http://www.r-project.org/)

3. spp package (http://compbio.med.harvard.edu/Supplements/ChIP-seq/)

4. R.matlab package (http://cran.r-project.org/web/packages/R.matlab/index.html)

5. UCSC Genome Browser Utilities (http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/)

Installation Instructions:

NOTE: These are installation/running instructions for 64-bit LINUX distributions. If you need executables for other platforms please contact Yun Zhu (zhuyun97@gmail.com).

2.1 Download package

Use `tar –zxvf epitensor.tar.gz` to unzip the package.

If separate files are downloaded (i.e. epitensor.tar.gz01…epitensor.tar.gz11), put all the 11 files in a single folder and use the following command to unzip the package:

```
cat epitensor.tar.gz* | tar -xzvf -
```

2.2 MCR Installation

In order to run the EpiTensor code and/or any MATLAB compiled code, you will need the MATLAB runtime library. Please only use the MCR version referenced in this README. This version of the executable was compiled using MCR V81 which is equivalent to R2013a release. You can download the MCR here http://www.mathworks.com/supportfiles/MCR_Runtime/R2013a/MCR_R2013a_glnxa64_installer.zip

If you haven't installed the MCR, you MUST do that using this command

1) Extract the contents of MCR_R2013a_glnxa64_installer.zip using the following command:

```
unzip MCR_R2013a_glnxa64_installer.zip
```

2) Run the MATLAB Runtime Installer script, from the directory where you unzipped the package file, by entering:

```
./install
```

The installer will prompt you to select the directory (<MCRROOT>) you want to install the MCR into, e.g. /home/yun/MATLAB/MATLAB_Compiler_Runtime/v81

Alternatively, you can install MCR non-interactively using the following commands:

```
./install -mode silent -agreeToLicense yes -destinationFolder <MCRROOT>
```

NOTE:

1) Make sure your installation directory has write permissions. The installation should go smoothly with the above command. However, if you are interested in other installation options you can consult http://www.mathworks.com/help/compiler/working-with-the-mcr.html#bs6mb58

2) You need to install the MCR ONLY once on the machine/cluster you plan to run MATLAB compiled code.

If you want to uninstall the MCR , follow this procedure:

Exit the application.

```
rm -rf <MCRROOT>
```

2.3 Setting paths

You need to set the following environment variables for the compiled MATLAB code to run correctly. These environment variables MUST be set before calling the epitensor executable or any other MATLAB compiled code.

You can add the following line to your .bashrc file:

`export MCRROOT=/usr/local/<MCRROOT>`, where <MCRROOT> is the root directory of MCR.

NOTE:

1) MCR also requires setting additional environment variables, as shown below. However, adding these lines to .bashrc may affect proper excecution of R program.

So please do NOT add the following lines to .bashrc. Instead, Epitensor will handle them for you.

DO NOT ADD THESE LINES

MCRROOT=<MCRROOT>/v81

LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${MCRROOT}/runtime/glnxa64

LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${MCRROOT}/bin/glnxa64

LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${MCRROOT}/sys/os/glnxa64

MCRJRE=${MCRROOT}/sys/java/jre/glnxa64/jre/lib/amd64

LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${MCRJRE}/native_threads

LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${MCRJRE}/server

LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${MCRJRE}

XAPPLRESDIR=${MCRROOT}/X11/app-defaults

export LD_LIBRARY_PATH

export XAPPLRESDIR

#############

2.4 R environment

1) If you are using Ubuntu or Debian, simply type in the following line

```
sudo apt-get install r-base r-dev
```

For more information, please check http://cran.r-project.org/bin/linux/ubuntu/README

If you are using Redhat, type in the following lines:

For EL5 or CentOS 5

su -c 'rpm -Uvh http://download.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm'

```
sudo yum update
```

```
sudo yum install R
```

For El6 or CentOS 6

```
su -c 'rpm -Uvh
http://download.fedoraproject.org/pub/epel/6/i386/epel-release-
6-8.noarch.rpm'

sudo yum update

sudo yum install R
```

You may also install R from source codes.

Check http://cran.r-project.org/doc/manuals/r-release/R-admin.html for details

2.5 spp package

1) Make sure that Boost C++ library (http://www.boost.org/) is already installed on your computer

Ubuntu users can simply use “apt-get install libboost-dev” to install the packages.

Alternatively, users can install from source files available on Boost C++ library website (http://www.boost.org/).

You may need to add the following lines to the .bashrc or .bash_profile files

CPLUS_INCLUDE_PATH=/path/to/your/include/folder:$CPLUS_INCLUDE_PATH

export CPLUS_INCLUDE_PATH

where “/path/your/include/folder/” is the installation path of Boost C++ library.

2) install.package(“caTools”) in R environment to install the “caTools” package.

3) Download spp package from http://compbio.med.harvard.edu/Supplements/ChIP-seq/

4) use "`R CMD INSTALL spp_1.10.tar.gz`" to install the package.

The command "library("spp")" should now work under R environment.

2.6 R.matlab package

Use install.packages("R.matlab") to install “R.matlab” package under R. The command "library("R.matlab")” should now work under R environment

2.7 UCSC Genome Browser Utilities

1) Download the binary codes from
http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/

2) No additional steps needed to compile and install the program

3) Need to add the base directory to your executable path

i.e. edit your ~/.bashrc file to include:

PATH=$PATH:/home/yun/software/kent/

The command wigToBigWig should now work from the command line.

2.8 Compile cpp file

Use ./epitensor/recompile.sh to compile the est_bkgd_lambda.c file.

3. Demos

Demo1

Goal: to illustrate the usage of `preprocessing.bash`

Input: one chip file (UCSD.H1.H3K4me3.SAK67.bam) and one input file (UCSD.H1.Input.DM219.bam) in bam format

Usage: ./demo1.bash

Output: coverage density profile (UCSD.H1.H3K4me3.SAK67.rdata) of the chip file (UCSD.H1.H3K4me3.SAK67.bam)

Demo2

Goal: to illustrate the usage of `epitensor.bash`

Input: a) datamatrixfile (datamatrix.txt) that specifies the location of 1-D epigenomic data

b) annofile that specifies the location of genome annotation (promoter, enhancer, exon, intron, and intergenic) file

Usage: ./demo2.bash

Output: promoter-promoter, promoter-enhancer, enhancer-enhancer interaction pairs in “out” path

Demo3

Goal: to illustrate how to identify interactions between cell-type-active promoter-promoter, promoter-enhancer, and enhancer-enhancer interactions

Input: a) promoter-promoter, promoter-enhancer, and enhancer-enhancer interactions identified in Demo 2

b) cell-type-active promoter, enhancer annotation file

Output: cell-type-active promoter-promoter, promoter-enhancer, and enhancer-enhancer interactions.

4. Command-line usage summary

a) `preprocessing.bash`

Description:

This command takes in an aligned read file in bam format and computes genome-wide coverge density

Usage:

```
preprocessing.bash -c chipfile -o covpath -r covfile -g genome [-i inputfile]
```

`chipfile` - aligned reads in bam format. If you have bed format files, please use the bedToBam function in bedtools (http://bedtools.readthedocs.org/en/latest/content/tools/bedtobam.html) to convert them to bam format.

`covpath` - the output path of coverage density file

`covfile` - the output coverage density file.

For example, if `covpath="/home/yun/output/"` and `covfile="H3K4me1.rdata"`, the output file is stored as "/home/yun/output/H3K4me1.rdata".

`genome` - This is the genome of organism for the input bam file. Currently, Epitensor only supports "hg19" genome and a test small-sized genome is also provided.

`inputfile` - aligned input/background reads in bam format. This is an optional parameter. For chipfiles without correspondent input files, this parameter can be left blank.

NOTE: Although `preprocessing.bash` handles only one chipfile at a time, it is easily parallelizable by submitting multiple jobs to a cluster, each job calling one `preprocessing.bash`

b) `epitensor.bash`

Description:

This command takes a set of pre-processed data files, performs tensor analysis, and outputs a set of promoter-promoter, promoter-enhancer, and enhancer-enhancer pairs.

Usage:

`epitensor.bash -f $datamatrixfile -h $annofile -o $outpath -w $workpath -g $genome -c $chr`

`datamatrixfile` - The data matrix file is a tab-delimited table in plain text (.txt) format. It can best be opened in MS Excel. It has the following format:

	assay_1	assay_2	…	assay_N
Cell_1				
Cell_2				
Cell_3				
…				
Cell_M				

An example training data matrix is as follows:

	H3K4me1	H3K4me3	H3K27me3	H3K27ac
hESC	/data/hESC/H3K4me1.rdata	/data/hESC/H3K4me3.rdata	/data/hESC/H3K27me3.rdata	/data/hESC/H3K27ac.rdata
TBL	/data/TBL/H3K4me1.rdata	/data/TBL/H3K4me3.rdata	/data/TBL/H3K27me3.rdata	/data/TBL/H3K27ac.rdata
MSC	/data/MSC/H3K4me1.rdata	/data/MSC/H3K4me3.rdata	/data/MSC/H3K27me3.rdata	/data/MSC/H3K27ac.rdata
NPC	/data/NPC/H3K4me1.rdata	/data/NPC/H3K4me3.rdata	/data/NPC/H3K27me3.rdata	/data/NPC/H3K27ac.rdata

NOTE: data must be pre-processed rdata format files. Raw data in bed or bam formats must be pre-processed before they can be used in the training process. See `preprocess.bash` for details

It is tediouly to manually create a data matrix, especially when the dimension is large. It is strongly recommended that the users use computer to generate such a file. An example code to generate a data matrix file is given in demo2.

`annofile` - This is the annotation file that specifies the location of promoter, enhancer, exon, intron, and intergenic regions.

`outpath` – This is the output path where promoter-promoter, promoter-enhancer, and enhancer-enhancer interactions are stored.

`workpath` - This is the temporary working directory. It is created by "epitensor.bash" and will be removed once the program ends.

`genome` - This is the genome of organism from which the data are obtained. Currently, epitensor only supports "hg19" genome and a test small-sized genome is also provided.

`chr` – This is the chromosome of the genome where the data are obtained. Currently, epitensor processes one chromosome at a time. If you want to process multiple chromosomes, you can run multiple epitensor.bash in parallel.

c) `extract_active_tss_active_tss_pairs`

Description:

This commands extracts cell-type-active promoter-promoter pairs from the promoter-promoter pairs identified by epitensor.bash

Usage:

```
extract_active_tss_active_tss_pairs.bash -a $all_tss_tss_file -t $anno_active_tss_file -o $active_tss_tss_file
```

`all_tss_tss_file` – promoter-promoter pairs from epitensor.bash

`anno_active_tss_file` – annotation file of cell-type-active promoters

`active_tss_tss_file` – cell-type-active promoter-promoter pairs

d) `extract_active_tss_active_enh_pairs`

Description:

This commands extracts cell-type-active promoter-enhancer pairs from the promoter-enhancer pairs identified by epitensor.bash

Usage:

```
extract_active_tss_active_enh_pairs.bash -a $all_tss_enh_file -t $anno_active_tss_file -e $anno_active_enh_file -o $active_tss_enh_file
```

`all_tss_enh_file` – promoter-enhancer pairs from epitensor.bash

`anno_active_tss_file` – annotation file of cell-type-active promoters

`anno_active_enh_file` – annotation file of cell-type-active enhancers

`active_tss_enh_file` – cell-type-active promoter-enhancer pairs

e) `extract_active_enh_active_enh_pairs`

Description:

This commands extracts cell-type-active enhancer-enhancer pairs from the enhancer-enhancer pairs identified by epitensor.bash

Usage:

```
extract_active_enh_active_enh_pairs.bash -a $all_enh_enh_file -
e $anno_active_enh_file -o $active_enh_enh_file
```

`all_enh_enh_file` – enhancer-enhancer pairs from epitensor.bash

`anno_active_enh_file` – annotation file of cell-type-active enhancers

`active_enh_enh_file` – cell-type-active enhancer-enhancer pairs