MuG - CHi-C Pipeline Documentation

Release 0.1

Pablo Acera

Feb 25, 2019

Table of Contents

CHAPTER 1

Requirements and Installation

1.1 Requirements

1.1.1 Software

- Python 2.7.12+
- R >=3.1.2
- bedtools
- perl
- HiCUP
- bwa
- samtools>1.3

1.1.2 Python Modules

- mg-tool-api
- pylint
- pytest
- rtree
- pyenv and pyenv-virtualenv
- rpy2
- matplotlib
- pandas
- rtree

- numpy
- scipy

1.1.3 R Modules

- argparser
- devtools
- Chicago

To Run runChicago.py and process_runChicago.py, the R script runChicago.R from https://bitbucket.org/chicagoTeam/chicago/src/ceffddda8ea392a1e84e4db9593f8fc35ac88048/chicagoTools/?at=master should be downloded and added to PATH.

1.2 Installation

Directly from GitHub:

```
git clone https://github.com/pabloacera/C-HiC.git
```

Using pip:

```
pip install git+https://github.com/pabloacera/C-HiC.git
```

Install R modules, use the following R code:

install.packages("argparser") install.packages("devtools") library(devtools) install_bitbucket("chicagoTeam/Chicago", subdir="Chicago")

CHAPTER 2

Full Installation

The following document is for the full installation of all software required by the C-HiC module and all programmes that it uses. The document has been written with Ubuntu Linux, although many of the commands are cross platform (*nix) complient.

If you already have certain packages installed feel free to skip over certain steps. Likewise the bin, lib and code directories are relative to the home dir; if this is not the case for your system then make the required changes when running these commands.

2.1 Setup the System Environment

```
sudo apt-get install -y make build-essential libssl-dev zlib1g-dev       \\
libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm libncurses5-dev \\
libncursesw5-dev xz-utils tk-dev unzip mcl libgtk2.0-dev r-base-core     \\
libcurl4-gnutls-dev python-rpy2 git libtbb2 pigz liblzma-dev libhdf5-dev \\
texlive-latex-base

cd ${HOME}
mkdir bin lib code
echo 'export PATH="${HOME}/bin:$PATH"' >> ~/.bash_profile
```

2.2 Setup pyenv and pyenv-virtualenv

This is required for managing the version of Python and the installation environment for the Python modules so that they can be installed in the user space.

```
git clone https://github.com/pyenv/pyenv.git ~/.pyenv
echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bash_profile
echo 'export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.bash_profile
echo 'eval "$(pyenv init -)"' >> ~/.bash_profile
```

(continues on next page)

(continued from previous page)

```

# Add the .bash_profile to your .bashrc file
echo 'source ~/.bash_profile"' >> ~/.bashrc

git clone https://github.com/pyenv/pyenv-virtualenv.git ${PYENV_ROOT}/plugins/pyenv-
↪virtualenv

pyenv install 2.7.12
pyenv virtualenv 2.7.12 C-HiC

# Python 3 environment required by iNPS
pyenv install 3.5.3
ln -s ${HOME}/.pyenv/versions/3.5.3/bin/python ${HOME}/bin/py3
```

2.3 Installation Process

2.3.1 bedtools and libspatialindex-dev

```
sudo apt-get install bedtools
sudo apt-get install libspatialindex-dev
```

2.3.2 Bowtie2 Aligner

```
cd ${HOME}/lib
wget --max-redirect 1 https://downloads.sourceforge.net/project/bowtie-bio/bowtie2/2.
↪3.4/bowtie2-2.3.4-linux-x86_64.zip
unzip bowtie2-2.3.4-linux-x86_64.zip
```

2.3.3 HiCUP

```
 cd ${HOME}/lib
 wget https://www.bioinformatics.babraham.ac.uk/projects/hicup/hicup_v0.6.1.tar.gz
 tar -xzf hicup_v0.6.1.tar.gz
 cd hicup_v0.6.1
 chmod a+x *
```

2.3.4 SAMtools

```
cd ${HOME}/lib
git clone https://github.com/samtools/htslib.git
cd htslib
autoheader
autoconf
./configure --prefix=${HOME}/lib/htslib
make
make install

```

(continues on next page)

(continued from previous page)

```
cd ${HOME}/lib
git clone https://github.com/samtools/samtools.git
cd samtools
autoheader
autoconf -Wno-syntax
./configure --prefix=${HOME}/lib/samtools
make
make install
```

2.3.5 Install CHiCAGO

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys␣
↪E298A3A825C0D65DFD57CBB651716619E084DAB9
sudo add-apt-repository 'deb [arch=amd64,i386] https://cran.rstudio.com/bin/linux/
↪ubuntu xenial/'
sudo apt-get update -qq
sudo apt-get install r-base-core
sudo apt-get install python-rpy2


cd ${HOME}/lib
sudo apt-get install libtbb-dev
sudo apt-get install libssl-dev
cd ${HOME}/C-HiC/
echo "R_LIB=${HOME}/R" > ${HOME}/.Renviron
echo "options(repos = c(CRAN = 'http://mirrors.ebi.ac.uk/CRAN/'))" > ${HOME}/.Rprofile
echo ".libPaths('~/R')" >> ${HOME}/.Rprofile
echo 'message("Using library:", .libPaths()[1])' >> ${HOME}/.Rprofile
sudo Rscript CHiC/tool/scripts/install_packages.R

cd ${HOME}/C-HiC/CHiC/tool/scripts/
wget https://bitbucket.org/chicagoTeam/chicago/raw/
↪e288015f75d36c5367d1595e0ac8099f2ce82aa1/chicagoTools/runChicago.R
wget https://bitbucket.org/chicagoTeam/chicago/raw/
↪e288015f75d36c5367d1595e0ac8099f2ce82aa1/chicagoTools/bam2chicago.sh
wget https://bitbucket.org/chicagoTeam/chicago/raw/
↪e288015f75d36c5367d1595e0ac8099f2ce82aa1/chicagoTools/makeDesignFiles.py
chmod +x bam2chicago.sh
```

2.4 Setup the symlinks

```
cd ${HOME}/bin



ln -s ${HOME}/lib/hicup_v0.6.1/* ${HOME}/bin/

ln -s ${HOME}/lib/bowtie2-2.3.4-linux-x86_64/bowtie2 bowtie2
ln -s ${HOME}/lib/bowtie2-2.3.4-linux-x86_64/bowtie2-align-s bowtie2-align-s
ln -s ${HOME}/lib/bowtie2-2.3.4-linux-x86_64/bowtie2-align-l bowtie2-align-l
ln -s ${HOME}/lib/bowtie2-2.3.4-linux-x86_64/bowtie2-build bowtie2-build
ln -s ${HOME}/lib/bowtie2-2.3.4-linux-x86_64/bowtie2-build-s bowtie2-build-s
```

(continues on next page)

(continued from previous page)

```
ln -s ${HOME}/lib/bowtie2-2.3.4-linux-x86_64/bowtie2-build-l bowtie2-build-l
ln -s ${HOME}/lib/bowtie2-2.3.4-linux-x86_64/bowtie2-inspect bowtie2-inspect
ln -s ${HOME}/lib/bowtie2-2.3.4-linux-x86_64/bowtie2-inspect-s bowtie2-inspect-s
ln -s ${HOME}/lib/bowtie2-2.3.4-linux-x86_64/bowtie2-inspect-l bowtie2-inspect-l

ln -s ${HOME}/lib/htslib/bin/bgzip bgzip
ln -s ${HOME}/lib/htslib/bin/htsfile htsfile
ln -s ${HOME}/lib/htslib/bin/tabix tabix


ln -s ${HOME}/lib/samtools/bin/ace2sam ace2sam
ln -s ${HOME}/lib/samtools/bin/blast2sam.pl blast2sam.pl
ln -s ${HOME}/lib/samtools/bin/bowtie2sam.pl bowtie2sam.pl
ln -s ${HOME}/lib/samtools/bin/export2sam.pl export2sam.pl
ln -s ${HOME}/lib/samtools/bin/interpolate_sam.pl interpolate_sam.pl
ln -s ${HOME}/lib/samtools/bin/maq2sam-long maq2sam-long
ln -s ${HOME}/lib/samtools/bin/maq2sam-short maq2sam-short
ln -s ${HOME}/lib/samtools/bin/md5fa md5fa
ln -s ${HOME}/lib/samtools/bin/md5sum-lite md5sum-lite
ln -s ${HOME}/lib/samtools/bin/novo2sam.pl novo2sam.pl
ln -s ${HOME}/lib/samtools/bin/plot-bamstats plot-bamstats
ln -s ${HOME}/lib/samtools/bin/psl2sam.pl psl2sam.pl
ln -s ${HOME}/lib/samtools/bin/sam2vcf.pl sam2vcf.pl
ln -s ${HOME}/lib/samtools/bin/samtools samtools
ln -s ${HOME}/lib/samtools/bin/samtools.pl samtools.pl
ln -s ${HOME}/lib/samtools/bin/seq_cache_populate.pl seq_cache_populate.pl
ln -s ${HOME}/lib/samtools/bin/soap2sam.pl soap2sam.pl
ln -s ${HOME}/lib/samtools/bin/varfilter.py varfilter.py
ln -s ${HOME}/lib/samtools/bin/wgsim wgsim
ln -s ${HOME}/lib/samtools/bin/wgsim_eval.pl wgsim_eval.pl
ln -s ${HOME}/lib/samtools/bin/zoom2sam.pl zoom2sam.pl
```

2.5 Prepare the Python Environment

2.5.1 Install APIs and Pipelines

Checkout the code for the DM API and the C-HiC pipelines:

```
cd ${HOME}/code
pyenv activate C-HiC
pip install --upgrade setuptools pip
pip install git+https://github.com/Multiscale-Genomics/mg-dm-api.git
pip install git+https://github.com/Multiscale-Genomics/mg-tool-api.git
pip install git+https://github.com/Multiscale-Genomics/mg-process-fastq.git

git clone https://github.com/pabloacera/C-HiC.git
cd C-HiC
pip install -e .
pip install -r requirements.txt
pip install dill
```

CHAPTER 3

Pipelines

3.1 Map and parse CHi-C reads

This pipeline will take as input two fastq files, RE sites, the genome indexed with GEM and the same genome in FASTA file. This pipeline uses TADbit to map, filter and produce a bed file that will be used later on to produce bam file compatible with CHiCAGO algorithm. More information about filtering and mapping https://3dgenomes.github.io/TADbit/

3.1.1 Running from the command line

Parameters

config [str] Configuration JSON file

in_metadata [str] Location of input JSON metadata for files

out_metadata [str] Location of output JSON metadata for files

Returns

Wd [folders and files] path to the working directory where the output files are

Example

REQUIREMENT - Needs two fastq files single end, FASTA genome and bowtie2 indexed genome.

When running the pipeline on a local machine without COMPSs:

```
python process_hicup.py \
   --config tests/json/config_hicup.json \
   --in_metadata tests/json/input_hicup.json \
```

(continues on next page)

(continued from previous page)

```
   --out_metadata tests/json/output_hicup.json \
   --local
```

When using a local version of the [COMPS virtual machine](https://www.bsc.es/research-and-development/software-and-apps/software-list/comp-superscalar/):

```
runcompss                      \
   --lang=python               \
   --library_path=${HOME}/bin \
   --pythonpath=/<pyenv_virtenv_dir>/lib/python2.7/site-packages/ \
   --log_level=debug           \
   process_fastq2bed.py          \
      --config tests/json/config_hicup.json \
      --in_metadata tests/json/input_hicup.json \
      --out_metadata tests/json/output_hicup.json
```

3.1.2 Methods

class process_hicup.**process_hicup**(*configuration=None*)

This class run hicup tool which run hicup, doing the alignment and filtering of the reads and convert them into a BAM file.

run(*input_files*, *metadata*, *output_files*)

This is the main function that runs

Parameters

- **input_files** (*dict*) – fastq1 fastq2
- **metadata** (*dict*) –
- **output_files** (*dict*) –

  out_dir: str directory to write the output

Returns

- **results** (*bool*)
- **output_metadata** (*dict*)

3.2 Create CHiCAGO input RMAP

3.3 Create CHiCAGO input BAITMAP

3.4 Create CHiCAGO input Design files

This script use as input .rmap and .baitmap files and generate the Design files. NPerBin file (.npb): <baitID> <Total no. valid restriction fragments in distance bin 1> … <Total no. valid restriction fragments in distance bin N>, where the bins map within the “proximal” distance range from each bait (0; maxLBrownEst] and bin size is defined by the binsize parameter. NBaitsPerBin file (.nbpb): <otherEndID> <Total no. valid baits in distance bin 1> … <Total no. valid baits in distance bin N>, where the bins map within the “proximal” distance range from each other end (0; maxLBrownEst] and bin size is defined by the binsize parameter. Proximal Other End (ProxOE) file (.poe): <baitID> <otherEndID> <absolute distance> for all combinations of baits and other ends that map within the “proximal” distance range from each other (0; maxLBrownEst]. Data in each file is preceded by a comment line listing the input parameters used to generate them.

3.4.1 Running from the command line

Parameters

config [str] Configuration JSON file

in_metadata [str] Location of input JSON metadata for files

out_metadata [str] Location of output JSON metadata for files

Returns

"nbpb" : .nbpb file "npb" : .npb file "poe" : .poe file

Example

REQUIREMENT - Needs RMAP and BAITMAP files

When running the pipeline on a local machine without COMPSs:

```
python process_design.py \
   --config tests/json/config_design.json \
   --in_metadata tests/json/input_design.json \
   --out_metadata tests/json/output_design.json \
   --local
```

When using a local version of the [COMPS virtual machine](https://www.bsc.es/research-and-development/software-and-apps/software-list/comp-superscalar/):

```
runcompss                      \
   --lang=python               \
   --library_path=${HOME}/bin \
   --pythonpath=/<pyenv_virtenv_dir>/lib/python2.7/site-packages/ \
   --log_level=debug           \
   process_design.py          \
      --config tests/json/config_design.json \
      --in_metadata tests/json/input_design.json \
      --out_metadata tests/json/output_design.json
```

3.4.2 Methods

class process_design.**process_design**(*configuration=None*)

This class generates the Design files and chinput files, imput for CHiCAGO. Starting from rmap and baitmap and capture HiC BAM files.

run(*input_files*, *metadata*, *output_files*)

Main function to run the tools, MakeDesignFiles_Tool.py and bam2chicago_Tool.py

Parameters

- **input_files** (*dict*) – designDir: path to the folder with .rmap and .baitmap files rmapFile: path to the .rmap file baitmapFile: path to the .baitmap file bamFile: path to the capture HiC bamfiles
- **metadata** (*dict*) – input metadata
- **output_files** (*dict*) – outPrefixDesign : Path and name of the output prefix, recommend to be the same as rmap and baitmap files. sample_name: Path and name of the .chinput file

Returns

- *bool*
- *output_metadata*

3.5 Convert BAM file into chicago input files .chinput

3.6 Data normalization and peak calling

This pipeline runs the normalization of the data and call the real chomatine interactions

3.6.1 Running from the command line

Parameters

config [str] Configuration JSON file

in_metadata [str] Location of input JSON metadata for files

out_metadata [str] Location of output JSON metadata for files

Returns

output_dir: directory with all output folders and files

Example

REQUIREMENT - Needs a reference genome

- **Needs file with the capture sequences with FASTA format**
  - settings file
  - **design dir:** .rmap .baitmap .npb .nbpb .poe

When running the pipeline on a local machine without COMPSs:

```
python process_run_chicago.py \
   --config tests/json/config_chicago.json \
   --in_metadata tests/json/input_chicago.json \
   --out_metadata tests/json/output_chicago.json \
   --local
```

When using a local version of the [COMPS virtual machine](https://www.bsc.es/research-and-development/ software-and-apps/software-list/comp-superscalar/):

```
runcompss                       \
   --lang=python                \
   --library_path=${HOME}/bin \
   --pythonpath=/<pyenv_virtenv_dir>/lib/python2.7/site-packages/ \
   --log_level=debug            \
   process_runChicago.py           \
      --config tests/json/config_chicago.json \
      --in_metadata tests/json/input_chicago.json \
      --out_metadata tests/json/output_chicago.json
```

3.6.2 Methods

class process_run_chicago.**process_run_chicago**(*configuration=None*)

Function for processing capture Hi-C fastq files. Files are aligned, filtered and analysed for Cpature Hi-C peaks

run(*input_files*, *metadata*, *output_files*)

This main function that run the chicago pipeline with runChicago.R wrapper

Parameters

- **input_files** (*dict*) – location with the .chinput files. chinput_file: str in case there is one input file chinput_file: comma separated list in case there is more than one input file.
- **metadata** (*dict*) – Input metadata, str
- **output** (*dict*) – output file locations

Returns

- **output_files** (*dict*) – Folder location with the output files
- **output_metadata** (*dict*) – Output metadata for the associated files in output_files

3.7 Run the entire CHi-C pipeline

CHAPTER 4

Tools for processing fastq C-HiC files

4.1 Map and parser reads

4.1.1 hicup_tool

class `CHiC.tool.hicup_tool.`**`hicup`**(*configuration=None*)

Tool to run hicup, from fastq to bam files

`digest_genome`(*genome_name*, *re_enzyme*, *genome_loc*, *re_enzyme2*)

This function takes a genome and digest it using a restriction enzyme specified

Parameters

- **`genome_name`** (`str`) – name of the output genome
- **`re_enzyme`** (`str`) – name of the enzyme used to cut the genome format example A^GATCT,BglII .
- **`genome_loc`** (`str`) – location of the genome in FASTA format
- **`re_enzyme2`** (`str`) – Restriction site 2 refers to the second, optional (other DNA shearing techniques such as sonication may be used) enzymatic digestion. This restriction site does NOT form a Hi-C ligation junction. This is the restriction enzyme that is used when the Hi-C sonication protocol is not followed. Typically the sonication protocol is followed.

static get_hicup_params(*params*)

Function to handle to extraction of commandline parameters and formatting them for use with hicup

Parameters **`params`** (`dict`) –

--bowtie	Specify the path to Bowtie
--bowtie2	Specify the path to Bowtie 2
--config	Specify the configuration file
--digest	Specify the digest file listing restriction fragment coordinates

--example	Produce an example configuration file
--format	Specify FASTQ format Options: Sanger, Solexa_Illumina_1.0, Illumina_1.3, Illumina_1.5
--help	Print help message and exit
--index	Path to the relevant reference genome Bowtie/Bowtie2 indices
--keep	Keep intermediate pipeline files
--longest	Maximum allowable insert size (bps)
--nofill	Hi-C protocol did NOT include a fill-in of sticky ends prior to ligation step and therefore FASTQ reads shall be truncated at the Hi-C restriction enzyme cut site (if present) sequence is encountered
--outdir	Directory to write output files
--quiet	Suppress progress reports (except warnings)
--shortest	Minimum allowable insert size (bps)
--temp	Write intermediate files (i.e. all except summaryfiles and files generated by HiCUP Deduplicator) to a specified directory
--threads	Specify the number of threads, allowing simultaneous processing of multiple files
--version	Print the program version and exit
--zip	Compress output

Returns

Return type list

`hicup_alig_filt`(*params*, *genome_digest*, *genome_index*, *genome_loc*, *fastq1*, *fastq2*, *outdir_tar*)

This function aling the HiC read into a reference genome and filter them

Parameters

- **`bowtie2_loc`** –
- **`genome_index`** (*`str`*) – location of genome indexed with bowtie2
- **`digest_genome`** (*`str`*) – location of genome digested
- **`fastq1`** (*`str`*) – location of fastq2 file
- **`fastq2`** (*`str`*) – location of fastq2

Returns

Return type Bool

`hicup_alig_filt_runner`(***kwargs*)

This function runs the hicup_alig_filt

Parameters

- **`bowtie2_loc`** –
- **`genome_index`** (*`str`*) – location of genome indexed with bowtie2

- **digest_genome** (*str*) – location of genome digested
- **fastq1** (*str*) – location of fastq2 file
- **fastq2** (*str*) – location of fastq2

Returns

Return type Bool

run (*input_files*, *metadata*, *output_files*)
Function that runs and pass the parameters for all the functions

Parameters

- **input_files** (*dict*) –
- **metadata** (*dict*) –
- **output_files** (*dict*) –

untar_index (*genome_file_name*, *genome_idx*, *bt2_1_file*, *bt2_2_file*, *bt2_3_file*, *bt2_4_file*, *bt2_rev1_file*, *bt2_rev2_file*)
Extracts the Bowtie2 index files from the genome index tar file. :param genome_file_name: Location string of the genome fasta file :type genome_file_name: str :param genome_idx: Location of the Bowtie2 index file :type genome_idx: str :param bt2_1_file: Location of the <genome>.1.bt2 index file :type bt2_1_file: str :param bt2_2_file: Location of the <genome>.2.bt2 index file :type bt2_2_file: str :param bt2_3_file: Location of the <genome>.3.bt2 index file :type bt2_3_file: str :param bt2_4_file: Location of the <genome>.4.bt2 index file :type bt2_4_file: str :param bt2_rev1_file: Location of the <genome>.rev.1.bt2 index file :type bt2_rev1_file: str :param bt2_rev2_file: Location of the <genome>.rev.2.bt2 index file :type bt2_rev2_file: str

Returns Boolean indicating if the task was successful

Return type bool

4.2 Create CHiCAGO input files

4.2.1 makeRmap

4.2.2 makeBaitmap

4.2.3 makeDesignFiles

class CHiC.tool.makeDesignFiles.**makeDesignFilesTool** (*configuration=None*)
Tool for makeing the design files as part of the input for Chicago capture Hi-C

static get_design_params (*params*)
This function handle chicago parameters, selecting the given ones and passing to the command line.

makeDesignFiles (***kwargs*)
make the design files and store it in the specify design folder. It is a wrapper of makeDesignFiles.py

Parameters

- **designDir** (*str,*) – Path to the folder with the output files(recommended the same folder as .map and .baitmap files).
- **parameters** (*dict,*) – list of parameter already selected by get_makeDesignFiles_params().

Returns

- *bool*
- **outFilePrefix** (*str*) – writes the output files in the defined location

run (*input_files*, *input_metadata*, *output_files*)
The main function to run makeDesignFiles.

Parameters

- **input_files** (`dict`) – designDir : path to the designDir containin .rmap and .baitmap files
- **input_metadata** (`dict`) –
- **output_files** (`dict`) –

  outFilePrefix [path to the output folder and prefix name of files] example: "/folder1/folder2/prefixname". Recommended to use the path to designDir and the same prefix as .rmap and .baitmap

Returns

- **output_files** (*dict*) – List of location for the output files.
- **output_metadata** (*dict*) – List of matching metadata dict objects.

4.3 Convert bam files into chicago input

4.3.1 bam2chicago

4.4 Normalize data and call C-HiC peaks

4.4.1 run_chicago

class `CHiC.tool.run_chicago.`**ChicagoTool** (*configuration=None*)
tool for running the CHiCAGO algorithm

chicago (***kwargs*)
Run and annotate the Capture-HiC peaks. Chicago will create 4 folders under the outpu_prefix data : output_index.Rds –> chicago data saved on Rds format output_index_params.txt –> parameters used to run Chicago output_index.export_format –> chicago output in the chosen format diag_plots : 3 plots to assest the quality of the output (see CHicago Capture-HiC documentation for details) enrichment_data: files for the feature enrichment output (in case is used) examples: output_index_proxExamples.pdf: random chosen peaks showing interactions regions see http://regulatorygenomicsgroup.org/chicago for more information

Parameters

- **input_files** (`str ot comma separated list if there is more than one replicate`) –
- **output_prefix** (`str`) –
- **output_dir** (`str (whole path for the output)`) –
- **params** (`dict`) –

Returns writes the output files in the defined location

Return type bool

static get_chicago_params(*params*)

Function to handle to extraction of commandline parameters and formatting them for use in the aligner for BWA ALN

Parameters **params** (*dict*) –

Returns

Return type list

run(*input_files*, *input_metadata*, *output_files*)

The main function to run chicago for peak calling. The input files are .chinput and are transformed from BAM files using bam2chicago.sh input files could be just one file or a comma separated files from more than one biological replicate. Technical replicates should be pooled to one .chinput

Parameters

- **input_files** (*dict*) – list of .chinput files, or str with a single .chinput file
- **input_metadata** (*dict*) –
- **output_files** (*dict with the output path*) –

Returns

- **output_files** (*Dict*) – List of locations for the output files,
- **output_metadata** (*Dict*) – List of matching metadata dict objects

static untar_chinput(*chinput_tar*)

This function take as input the tar chinput

Parameters **chinput_tar** (*str*) – path to the tar file, the tar files should have the same prefix name as the tar file

Returns

Return type list of untar files

CHAPTER 5

Architectural Design Record

5.1 25-09-2018 handling_chr_header branch merge with master

This rmap_tool.py from this branch take the chromosome format from the used the reference genome and output a file with two columns, dictionary like with number of the chromosome and the name of the chromsome from the reference genome. example 1 chr1 2 chr2 3 chr3 ect…

This file is passed to the makeBaitmap.py script and generate the .batimap file with the corresponding chromsome name. This is necesary as the rtrees used in makeBaitmap.py needs an integer instead of “chr” or any other format.

5.2 15-10-2018 mm_mods_for_makebaitmaps branch merge with master

This branch contains some modifications from Mark to solve issues with pyCOMPSs regarding makeBaitmap.py tool

CHAPTER 6

License

Apache License Version 2.0, January 2004 http://www.apache.org/licenses/

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

   "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index