DCdoc Documentation

Release 0.30

DC

Jun 11, 2017

Contents

Cpipe is a comprehensive and user friendly ChIP-seq Analysis Pipeline written in Python.

See *Simpest config* for a quick view of how to use this pipeline.

CHAPTER 1

Manual

About the project

Purpose

- ChIP-seq experiment has been a mature and wide-spread technique for detecting the TF and histone modification distribution from the genome scale.
- Along with the popularity of the technique and the increasingly huge number of highthroughput datasets, it may be confusing for biologists to get a quick and easy access to understand their biological meaning, and the same important thing is the unbiased judgement of the data quality.So, it's necessary for us establishing a universal and easy-to-use ChIP-seq data analysis pipeline for biologists.
- For bioinformaticians, you may find this may be the most extensible and flexible ChIP-seq integrated python packages ever. It provides various genomic data format support and high-rated analysis toolbox. Any bug report is welcome.

Goal

There are two layers for pipeline goal.This is part of the Cistrome project

1.command line

```
The goal is to simply input the sequence data files up to input format
support and fill in the customized table of experiment descriptions,
output the desired format of DC and QC report
```

2.web server

```
The ChIP-seq pipeline could be incorporated into the Cistrome
```

Features

This page is intended for:

Developers: in order to be sure they are developing the right project that fulfills requirements provided in this document.

Users: in order to get familiar with the idea of the project and suggest other features that would make it even more functional.

Cpipe should include the following features.

Supported formats

Cpipe should support the following format of `Raw Data` as input:

Format	Type	Steps
FASTQ	Seq	
BED	Mapped	Skip mapping
BAM	Mapped	Skip mapping

Cpipe may **not** support the following format in current version:

Format	Type	Solution
SRA	Seq	Use SRA Toolkit to convert to FASTQ format
BED	Summit	
BED	Peak	
wig	Profile	
Bigwig	Profile	

Data preprocession

Convert the raw sequencing data into intervals and profiles.

- Use Bowtie for tag alignment (mapping)
- Use MACS2 for peak calling

Correlation

Focus on the visulization of similarity between replicates.

- Draw the venn diagram for peaks if there're less than 3 replicates (treatment or control)

Association Study

Focus on association between intervals (result of peak calling) and traits like genome annotation.

- CEAS: Annotate the given intervals and scores with genome features
- Conservation Plot: Calculates the PhastCons scores in several intervals sets

Motif

analysis the motif of the binding sites.

Quality control

Based on Chip-seq pipeline and Cistrome DC database, QC program will generate a comprehensive quality control report about a particular dataset as well as the relative result compared to the whole DC database.

- Basic information: Species, Cell Type, Tissue Origin, Cell line, Factor, Experiment, Platform, Treatment and Control.
- Reads Genomic Mapping QC measurement: QC of raw sequence data with FastQC, FastQC score distribution, Basic mapping QC statistics, Mappable reads ratio, Mappable Redundant rate.
- Peak calling QC measurement: Peak calling summary, High confident Peak, Peaks overlapped with DHS(Dnase Hypersensitivity sites), Velcro ratio(human only), Profile correlation within union peak regions, Peaks overlap between Replicates.
- Functional Genomic QC measurement: Peak Height distribution, Meta Gene distribution, Peak conservation score, Motif QCmeasurement analysis.

This page is intended for:

Developers: in order to make sure they're using the right format of data and right version of tool to test

Users: in order to know where they should go to download these data and tools

Data

Built-in Data

The Cpipe package includes all the build-in data for hg19 and mm9. For other species, you may need to download these data from data source or custom it yourself.

Data Name	Used by	Data Source	Format
Chromesome length	samtools	UCSC table browser	2-column
Chromesome length	CEAS	–	–
Genome backgroud annotation	CEAS	CEAS site	sqlite3
DHS region	bedtools	Custom	BED
Velcro region	bedtools	Custom	BED
Motif database	MDSeqPos	MDSeqPos site	xml
FastQC result database	QCreport	Custom	bed
Data summary database	QCreport	Custom	bed

External Data

Some data are too large to be included by the pipeline package, so you need to download these data from data source.

Data Name	Used by	Data Source	Format
Bowtie pre-built index	Bowtie	Bowtie site	ebwt
Conservation profile	Conservation Plot	Cistrome site	Bigwig

Tools

Built-in Tools

Built-in tools are the scripts that can be run from command-line independently when you have installed the Cpipe package.

Tool Name	Modified from
Venn Diagram	
Conservation Plot	
Correlation plot	bigwig_correlation
bamtofastq	
wigTobigwiggle	
RegPotential	
sample_contamination	

External Tools

External Tools are the tools invoked by Cpipe by their path.

Tool Name	Download source	Version
FastQC		
R		
Cython		
MACS2	MACS site	2.0.10 20120605
CEAS	CEAS site	0.9.9.7
bedtools	bedtools site	v2.16.2
pybedtools		
samtools	SAMtools site	0.1.17
Bowtie	Bowtie site	0.12.8
bedGraphToBigWig	UCSC utilities	v4
FastQC	FastQC site	v0.10.1
pdfTeX	pdfTex site	v1.40.10
IGV		

Config file

Synopsis

[meta]

Lists all the meta-data of current workflow.

Consist of the following options:

dataset.ID

The name for the dataset, which will be the value of *${DatasetID}*

Limit: a string (1) consist of `numbers`, `alphabets` or `'_'` (2) shorter than 20 characters

species

The name of species, written to the QCreport and log

Limit: a string (1) consist of `numbers`, `alphabets` or `'_'` (2) shorter than 20 characters

assembly
The assembly version, written to the QCreport and log

Limit: a string (1) consist of `numbers`, `alphabets` (2) shorter than 10 characters

treatment
The paths of treatment files

Limit: absolute or relative `path` of files in *supported formats*

control
The paths of treatment files

Limit: absolute or relative `path` of files in *supported formats*

[ext]
The external data and external tools to use. Read *External Data* and *External Tools* for a full explanation.

[steps]
meta

Simpest config

Here is one of the simpest Cpipe workflow you can make.

```
1 [meta]
2 dataset.ID = mydata_ctcf
3 species = human
4 assembly = hg19
5
6 treatment = .../GSM489301.fastq
7
8 [ext]
9 directory = .../Cpipe/data
```

Use your own path of *Raw Data* to replace the Line 6. And use the path of the directory used to store *External Data* to replace Line 9.

When saved to `hello_cpipe.conf`, this config file can construct a powerful pipeline via:

```
$ Cpipe hello_cpipe.conf
```

When it finished about 2 hours later, you will get *Processed Data* and a *Final PDF Report*.

Examples about replicates

For example, there are five raw files. Three of them are replicates for `treatment` and two of two for `control`.

The file names may look like:

```
demo_treat1.fastq
demo_treat2.fastq
demo_treat3.fastq
demo_control1.fastq
demo_control2.fastq
```

Then you can write the *[meta]* section like this:

```
1 [meta]
2 dataset.ID = demo_replicate
3 # species = human
4 # assembly = hg19
5 treatment.1 = demo_treat1.fastq
6 treatment.2 = demo_treat2.fastq
7 treatment.3 = demo_treat3.fastq
8 control.1 = demo_control1.fastq
9 control.2 = demo_control2.fastq
```

Replace the commented in Line 2, Line 3 and Line 4 and complete other sections. Then load it with Cpipe.

For the notation of output files, the *${DatasetID}* will be demo_replicate. The *${treat_rep}* will be 1, 2 and 3. The *${control_rep}* will be 1 and 2.

Output & Temporary files

This page is intended for:

Developers: in order to make sure they have a consistent naming convention and can find each file easily

Users: in order to know what each output file represents

Through the pipeline, several temporary files will be generated, some of them are only used for settings and transitions, others for continuing the next step, the rest for publishing and interpreting a biological story.Below is three sections of tables for universal name rules.

Note: For data which has not been published on GEO use factor name plus your favorate number to replace the GSMID below.

Notation

${DatasetID}
: The value of dataset.id option in *[meta]* section

${treat_rep}
: The suffix of *treatment* option in *[meta]* section

${control_rep}
: The suffix of *control* option in *[meta]* section

Temporary files

Name	Content	Tool used
${DatasetID}_treat_rep${treat_rep}.sam	mapping result	*External Tools*
${DatasetID}_control_rep${control_rep}.sam	mapping result	*External Tools*
${DatasetID}.conf	configuration	Main program
${DatasetID}_bedtools_dhs.txt	DHS peaks intersection	*External Tools*
${DatasetID}_cor.R	correlation plot code	Buit-in tools
${DatasetID}_seqpos.zip	Motif analysis	*External Tools*
${DatasetID}_QC.tex	QC report code	pdftex
${DatasetID}_mappable_ratio.pdf	Mapping QC result	R
${DatasetID}_fastqc_score_distribution.pdf	Raw data QC	R
${DatasetID}_peak_distribution.pdf	Peak calling QC	R
${DatasetID}_velcro_ratio.pdf	Peak calling QC	R
${DatasetID}_peak_overlap_DHS.pdf	Peak calling QC	R

Output result

Name	Content	Tool used
Folder	containing all results	Main Program
GSMIDlog	log	
GSMID_control_repnumber.bam	mapping result	*External Tools*
GSMID_treat_repnumber.bam	mapping result	
GSMID_repnumber_peaks.bed	Peak calling	*MACS2*
GSMID_bedtools_dhs.txt	DHS peaks intersection	*BEDtools*
GSMID_cor.R	correlation plot code	*Built-in tools*
GSMID_seqpos.zip	Motif analysis	*MDSeqpos*
GSMID_QC.tex	QC report code	**pdftex_**
GSMID_ceas.xls	CEAS	**CEAS_**
GSMID_conserv.png	Phascon score plot	*Built-in tools*
GSMID_conserv.R	Phascon score	*Built-in tools*

Final PDF Report

Provide the overall report of the whole pipeline for viewing general result.

Name	Content	Tool used
GSMID_ceas_combined.pdf	Cistron annotation	CEAS
GSMID_QC.pdf	All quality control measurements	Main program

CHAPTER 2

API Documentation

Documentation for every chlin API.

API Documentation

Controllers of QC

class chilin.qc.**QC_Controller**(*template=''*)
All the class in the module derives from this class

check()
Check whether the quality of the dataset is ok.

render(*template=None*)
Generate the latex code for current section.

run()
Run some QC tools or do some time-costing statistics

class chilin.qc.**RawQC**

class chilin.qc.**MappingQC**

class chilin.qc.**PeakcallingQC**

class chilin.qc.**AnnotationQC**

CHAPTER 3

Others

- genindex

Python Module Index

C

Index