# The ChIP-Seq Distillery

*Beta Version 01032016*

# User Manual

February 2016

**Victor Hanson-Smith**

Victor.hanson-smith@ucsf.edu

victorhansonsmith@gmail.com

# Introduction

Chromatin immunoprecipitation followed by DNA sequencing, commonly known as ChIP-seq, is a popular experimental technique to determine genomic locations that are enriched for binding by a particular DNA-binding protein. A common experimental approach is to perform ChIP-seq in different conditions – perhaps in different environmental conditions, or different species – and then compare the genomic location bound by a DNA-binding protein of interest.

The ChIP-Seq Distillery is a software pipeline that automates the multi-step protocol of aligning reads to a genome, calling peaks, and then comparing those peaks between replicates or species. The Distillery combines several popular software tools into a single unified pipeline.

Table of Contents

The ChIP-Seq Distillery 1

User Manual 1

Introduction 2

Download and Install 4

Not Designed for Laptops 4

Prerequisites 4

Install 4

Usage 6

Command: 6

Required Parameters: 6

Optional Parameters: 7

Input Files 9

Configuration Files 9

Genomes 10

Genome Feature Files 10

Synteny Maps (i.e. Pillars Files) 10

The Analysis Pipeline 11

Output Files 12

# Download and Install

The source code for the ChIP-Seq Distillery is available as a repository on GitHub, available at the following URL: https://github.com/vhsvhs/SeqTools

## Not Designed for Laptops

The Distillery is designed to run on clusters or supercomputers running the Linux, Unix, or OSX operating systems. Computer analysis of ChIP-Seq data can run for many hours and sometimes days; it requires large amounts of computer memory, and this analysis may be inappropriate for your laptop computer.

## Prerequisites

The Distillery is written in **Python** and **R**. It makes use of functions in three commonly-used Python libraries: **scipy**, **numpy**, and **matplotlib**. You will need to install these libraries separately.

* SciPy: <http://www.scipy.org>
* NumPy: <http://www.numpy.org>
* Matplotlib: <http://matplotlib.org>

You will also need to install software for aligning reads to genomes (Bowtie), and for inferring the locations of binding peaks (MACS2):

* Bowtie 2: <http://bowtie-bio.sourceforge.net/bowtie2/index.shtml>
* MACS2: <https://github.com/taoliu/MACS/>

## Install

The Distillery itself does not need to be installed, *per se*, but you will need to download its source code as follows:

1. Navigate to the folder where you would like to install the ChIP-Seq Distillery. For example:

cd ~/Applications

1. Clone the GitHub repository containing the most-recent source code:

git clone https://github.com/vhsvhs/SeqTools

1. At this point, you can launch the distillery by running the python script located at PATH/after\_illumina/run\_distill.py, where PATH is the folder in which you cloned the GitHub repository. For example:

Python ~/Applications/after\_illumina/run\_distill.py

Note for UCSF Johnson Lab: The software is already installed on the lab server. It can be invoked as a Python script at the following filepath:

/common/REPOSITORY/SeqTools/after\_illumina/run\_distill.py

# Usage

The ChIP-Seq Distillery (CSD) is a Python script that you run from the command-line. The program may require several hours to finish, depending on the size of your ChIP-seq project. During its run, the software will print useful updates to the terminal.

## Command:

**python run\_distill.py <parameters>**

## Required Parameters:

**--configpath PATH**

PATH is the filepath to a configuration file for your analysis. See the section titled “Configuration Files” for more information.

**--datadir PATH**

PATH is the path to a folder containing your FASTQ reads. A common workflow is to store FASTQ files in a central location, and run analysis (such as the ChIP-Seq Distillery) in a different folder.

**--outdir PATH**

PATH is the path to a folder where output files will be created. **NOTE for June 2015:** It is recommended that you set your output directory PATH to the value “./”, navigate to this folder, and run the ChIP-Seq Distillery from within the output folder.

**--project\_name NAME**

NAME is an arbitrary title for your project. It can be any combination of letters, numbers, and punctuation, but it cannot contain any spaces. For example, a good project name is to combine the name of your tagged gene with the date: “MyGene.June.26.2015”. The project name is used as a unique keyword when generating some output files.

**--pillarspath PATH**

PATH is the filepath to a synteny map (i.e. a pillars file).

## Optional Parameters:

**--dbpath PATH**

PATH is the filepath of the SQL database that will be created (or restored) for this analysis. The default value is NAME.db, where NAME is the project name specified by the parameter “—project\_name.”

**--jump NUMBER**

NUMBER is a stage in the software pipeline. The software will launch directly into the desired stage, skipping all previous stages. This options requires that you specify a database path, using the parameter –dbpath, and the database comes from an analysis that previously succeeded in the stages prior to NUMBER. See the section titled “Pipeline Stages” for more information.

**--stop NUMBER**

NUMBER is a stage in the software pipeline. The software will stop when it reaches stage NUMBER. See the section titled “Pipeline Stages” for more information. By combining the parameters –start and –stop, you can run or re-run specific parts of the ChIP-Seq distillery pipeline. The NUMBER for –stop should always be larger than the NUMBER for –start, otherwise no analysis will occur at all.

**--use\_mpi BOOL**

BOOL can be 0 or 1. If 0, then the distillery will run serially. If 1, then the distillery will use MPI to run some stages in parallel.

**--mpinp NCORES**

NCORES is a number specifying the number of MPI processes to launch. The default value is 4. This parameter is only valid if “–use\_mpi 1” is specified.

**--mismatch\_thresh VALUE**

VALUE is an integer specifying the maximum allowed mismatches in a sequenced read. All reads with mismatches greater than VALUE are ignored. If VALUE equals -1, then the threshold is removed and all reads will be considered regardless of their mismatch level. The default value is -1.

**--chrom\_filter KEYWORD**

Restricts read-aligning and peak calling only to those chromosomes whose names include the KEYWORD.

**--bowtie2 PATH**

PATH is the executable path of Bowtie2. The default is “bowtie2 –p 4”.

**--samtools PATH**

PATH is the executable path of Samtools. The default is “samtools”.

**--macs2 PATH**

PATH is the executable path of Macs2. The default value is “macs2”.

**--gcb PATH**

PATH is the executable of BedTools Genome Coverage Bed. The default is “genomeCoverageBed”

**--use\_multialign**

No value required. If this parameter is specified, then reads mapped to multiple locations in the genome will be used in subsequent analysis. This option is disabled by default; i.e., multi-mapped reads are ignored by default.

**--minqval VALUE**

VALUE is the Q-value cutoff to call significant peaks in the MACS2 software. The default behavior is to use the default Q-value cutoff in MACS2, which is 0.05 in MACS2 version 2.

**--practice\_mode**

No value required. If this parameter is specified, then the software will not launch compute-intensive applications, including Bowtie2 and Macs2. Rather, the software will read input files, and only write control scripts. This feature is intended for developer use only.

# Input Files

## Configuration Files

The configuration file includes information about the FASTQ read files in your analysis, the genomes used for mapping those reads, the genome annotations for mapping peaks to genes, and the experimental design that pairs tagged reads to untagged reads. A configuration file contains lines starting with six different keywords:

**GENOMEPATH <genome name> <genome path>**

* <genome name> is an arbitrary name for the genome. This name is referenced by other lines in the configuration file, so it is recommend that you pick a short and meaningful name. For example, the genome for species *Saccharomyces cerevisiae* could be named, simply, *Scer*. The only constraint on genome name is that it must be unique and it must be a single word, i.e. no spaces.
* <genome path> is the fileprefix to a FASTA-formatted genome file.

**BOWTIEINDEX <genome name> <index prefix>**

* <genome name> is an arbitrary name for the genome. See the notes about <genome name> above, for the keyword GENOMEPATH
* <index prefix> is the fileprefix to the indexed genome files, which typically end in “.1.bt2”, “.2.bt2”, etc. For example, if your bowtie index files are located /Documents/genomes/cerevisiae.1.bt2, /Documents/genomes/cerevisiae.2.bt2, etc., then your value for <index prefix> should be /Documents/genomes/cerevisiae

**GFF <genome name> <GFF path>**

* <genome name> is the name of genome, and should match the name used in GENOMEPATH and BOWTIEINDEX lines.
* <GFF path> is the filepath to a genome feature file, which typically ends in the suffix “.gff”.

**INTERGENIC <genome name> <INTERGENIC path>**

* <genome name> is the name of genome, and should match the name used in GENOMEPATH and BOWTIEINDEX lines.
* <GFF path> is the filepath to a genome feature file, which typically ends in the suffix “.gff”.

**READS <unique library name> <FASTQ path> <genome name> <gene name> <tagged?> <condition> <note>**

* <unique library name> is a arbitrary, but unique, name for this library of reads.
* <FASTQ path> is a filepath to a FASTQ file.
* <genome name> is the name of a genome used previously in a GENOMEPATH and BOWTIEINDEX line. The reads in the fastq file will be mapped to this genome.
* <gene name> is the name of the tagged gene in the reads. This can be any arbitrary value, including “none” or “n/a” for control reads
* <tagged?> can be “YES” or “NO”, indicating if the reads in the FASTQ file correspond to a tagged ChIP experiment or not.
* <condition> is an arbitrary word indicating the growth conditions, genomic background, or other conditions unique to this experiment. A value for condition is required, but is not currently used by the pipeline.
* <note> is optional, and can be as many words as you like, but it must come after the six values listed above.

**HYBRID <library name #1> <library name #2>**

* HYBRID lines are optional, and only necessary if your analysis includes READS that can be mapped to multiple species.
* <library name #1> and <library name #2> are the unique read names of two READS lines, listed above. The HYBRID line indicates that these libraries share the same FASTQ file, but came from a hybrid species. The two READS lines should have different values for their <genome name>.

**EXPERIMENT <unique experiment name> : <tagged library name #1> <untagged library name #2>**

* EXPERIMENT lines pair tagged with untagged ChIP reads.
* <unique experiment name> is an arbitrary, but unique, name for this experimental pair
* <library name #1> is the library name of a READS line whose value for <tagged?> is YES>
* <library name #2> is the library name of a READS line whose value for <tagged?> is NO.

**COMPARE <unique comparison name> : <target 1> <target 2> <target 3>**

* COMPARE lines indicate experiments whose peak and fold-enrichment values should be compared. You can include two or three targets.
* <unique comparison name> is an arbitrary, but unique, name for this comparison
* <target 1> is the experiment name of an EXPERIMENT line, or the comparison name of a previously-listed COMPARE line.
* <target 2> is the experiment name of an EXPERIMENT line, or the comparison name of a previously-listed COMPARE line.
* <target 3> is optional, and is the experiment name of an EXPERIMENT line, or the comparison name of a previously-listed COMPARE line.

## Bowtie Indexes

Indexed genomes are imported into the ChIP-Seq Distillery by including BOWTIEINDEX lines in the configuration file. The indexes should be indexed Bowtie2 files.

## Genome Feature Files

Genome feature files are imported into the software by including GFF lines in the configuration file. The feature file must be in the GFF version 3 format.

## Synteny Maps (i.e. Pillars Files)

The synteny map specifies gene orthologs between different species. The synteny map is imported into the software by specifying the –pillarspath parameter when invoking the ChIP-Seq Distillery.

# Stages in the Analysis Pipeline

The stages in the ChIP-Seq Distillery occur in the order listed below. Stages can be repeated by using the parameters –start and –stop (see the section titled “Usage” for more information).

|  |  |
| --- | --- |
| **Stage** | **Action** |
| 1 | Map reads to genome (run Bowtie) |
| 1.1 | Check the Bowtie results |
| 2 | Extract reads from the Bowtie output |
| 2.3 | (Hybrid Experiments Only) Find reads that are unique to one, but not both, of hybrid pairs. |
| 2.31 | Create read histograms (PDF output) |
| 2.4 | Write filtered SAM files containing only those reads that were extracted in stage 2. |
| 3 | Write a sorted BAM file for every SAM file created in stage 2.4 |
| 3.1 | Verify that BAM files were correctly created. **If the BAM files are OK, then the original SAM files will be deleted.** |
| 3.2 | Make bedgraph files of read densities (This will run BedTools Genome Coverage Bed) |
| 3.3 | Check the bedgraph files created in step 3.2 |
| 4 | Peak Calling (This will run MACS2) |
| 4.1 | Check the MACS2 results |
| 5 | Calculate fold enrichment (run MACS bdgcmp) |
| 5.1 | Check for existence of fold-enrichment output from MACS2. |
| 6 | Create WIG files (for MochiView) for raw read densities and fold-enrichment. |
| 6.2 | Check the WIG files created in stage 6. |
| 7 | Setup the visualization and comparison of experiments. |
| 8 | Compare the experiments, and write output in the form of Excel spreadsheets, PDFs, and PNGs. |
| 9 | Write HTML output (NOTE for June 2015: this stage is still in development) |

# Output Files

The ChIP-Seq distillery produces many different output files at various stages throughout the pipeline. The output is written into the output folder specified by the parameter “—outdir”. The most salient output files are listed below, although additional files can be found in the output directory.

**Experimental Peak List (Excel)**

* Every EXPERIMENT entry in the configuration file will result in an Excel spreadsheet titled <name>\_peaks.xls. This file is written by MACS2.

**Comparison of Fold Enrichment (Excel)**

* Every COMPARE entry in the configuration file will result in an Excel spreadsheet titled <comparison name>.enrich.xls. This file includes one row for every gene, and indicates the fold-enrichment and summit scores for every experiment included in the COMPARE.

**Comparison of Fold Enrichment (PNG)**

* Every COMPARE entry in the configuration file will result in a PNG figure titled <comparison name>.maxfe.png. This file correlates the maximum fold enrichment for every gene between the experiments in the COMPARE line. Genes with summits are also indicated in this figure.

**Irreproducible Discovery Rate (PDF)**

* Every COMPARE entry in the configuration file will result in a PDF figure showing the IDR between the experiments in the COMPARE line. This file is titled <comparison name>.enrich.idr.pdf. Additionally, the R script used to generate the figure is written to the file <comparison name>.enrich.idr.cran.

**Project Database (SQL)**

* The ChIP-Seq Distillery stores all imported files and data in a SQL3 database at the filepath specified by the parameter –dbpath. This file can be very large, often several gigabytes in size.