Sequence Motif Discovery Manual

Introduction 2
- About this Pipeline 2
- Workflow 3

Packages 4
- FastQC 4
- Cutadapt 4
- STAR Aligner 4
- SAMtools 5
- barcodecollapsepe.py 5
- Anaconda 5
- deepTools 5
- PEAKachu 6
- MEME Suite 6
- Cthreepo 6
- RNA Centric Annotation System 7

Pipeline Instructions 7
- Required Input Files 7
- Usage 8

Pipeline Output 9
- Directories Structure 9
- Overall Standard Output and Error Information 10
- FastQC 11
- Cutadapt 11
- STAR Aligner 11
- Samtools 12
- barcodecollapsepe.py 12
- deepTools 12
- plotFingerprint 13
- plotCorrelation 13
- PEAKachu 14
- MEME Suite 15
- Downstream Analysis 15

Pitfalls and Limitations 16

Introduction

About this Pipeline

This pipeline utilizes enhanced Cross-linking immunoprecipitation coupled with high-throughput (eCLIP) data to discover possible motifs in the human genome. CLIP-seq data is often purposed to resolve binding sites in a given target gene sequence, allowing the discovery of sequence motifs recognized by specific RNA-binding proteins (RBPs) and lending insights into binding functionality. The eCLIP data, a set of RNA elements recognized by RBPs, are collected from the Encyclopedia of DNA Elements (ENCODE) project. eCLIP data for each experiment on various RBPs is provided on the ENCODE portal and are freely available.

Here we modified the procedures from Blue et al.[1] paper and the Galaxy Project[2] for processing eCLIP data, making a tailored experimental reference bash script that runs on the supercomputer Bridges-2 from PSC[3]. The run of the pipeline in this manual incorporates eCLIP data in HepG2 dataset on the TAF15 target.

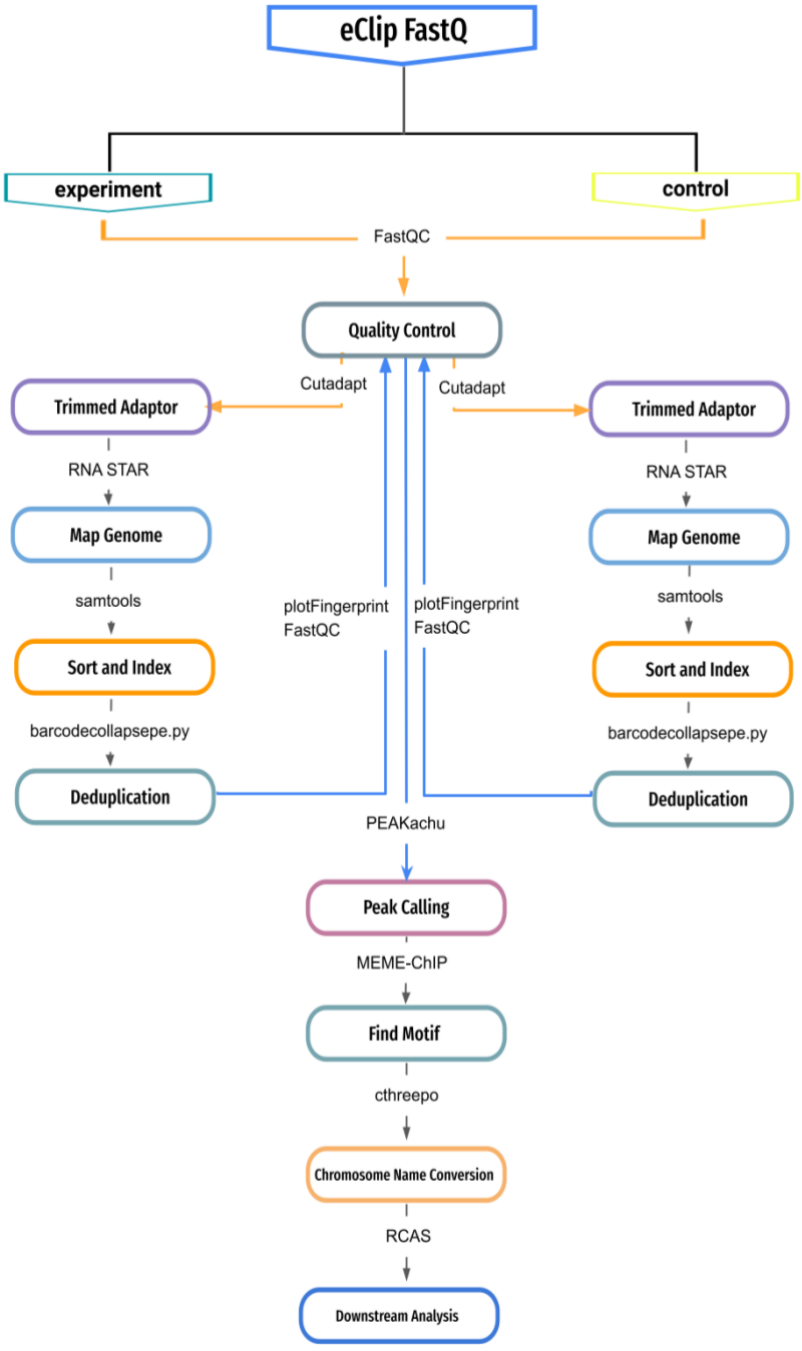
Our workflow is composed of mainly four steps and they are explained briefly through the packages we used. *FastQC* is a quality check tool and was used to detect bias and duplications in eCLIP-seq experiment data. *RNA STAR* was used for aligning reads to the reference genomes. The resulting alignment files were used for peak calling using *PEAKachu*. Lastly, *MEME-ChIP* performs comprehensive motif analysis. This package was used to discover motifs on sequences by locating ChIP-seq peaks and identifying the center of the peaks.

The whole pipeline is constructed based on *bash* scripts. An additional downstream analysis is also provided along with a modified *RNA Centric Annotation System* R script for specific usage. This additional part generates functional annotation and GO term enrichment analysis for

[1] Blue, S.M., Yee, B.A., Pratt, G.A. et al. Transcriptome-wide identification of RNA-binding protein binding sites using seCLIP-seq. *Nat Protoc* (2022).
[2] https://training.galaxyproject.org/training-material/topics/transcriptomics/tutorials/clipseq/tutorial.html
[3] https://www.psc.edu/resources/bridges-2/

further biology research. The downstream analysis is not supported on bridges.

Workflow



Packages

Note: For the list of pre-installed packages in Bridges-2, please refer to the website: Software Installed on PSC Systems. For parameters, please refer to the “Running Pipeline: Parameters section” of this user manual.

FastQC

FastQC package is a pre-installed tool in Bridges-2. This package is used for evaluating the quality of the raw sequencing data file by measuring the duplication level of reads with different lengths. PCR duplicates need to be erased before the following steps in the pipeline as they can give false positive results to the analysis.

For usage on Bridges-2, refer to: FastQC | PSC
For documentations, refer to: FastQC

Cutadapt

Cutadapt package is a pre-installed tool in Bridges-2, and is used for finding and removing adapter sequences, primers, poly-A tails and other types of unwanted sequence from your high-throughput sequencing reads.

For usage on Bridges-2, refer to: Cutadapt | PSC
For documentations, refer to: Cutadapt

STAR Aligner

STAR Aligner package is a pre-installed tool in Bridges-2, which is used for mapping reads from the spliced transcripts alignment to a reference with support for splice-junction and fusion read detection.

For usage on Bridges-2, refer to: SAMtools | PSC
For documentations, refer to: SAMtools

SAMtools

SAMtools package is a pre-installed tool in Bridges-2, which provides various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing and generating alignments in a per-position format.

For usage on Bridges-2, refer to: STAR Aligner | PSC
For documentations, refer to: STAR Aligner

barcodecollapsepe.py

barcodecollapsepe.py is a python script from yeo's lab Github homepage used for deduplication of pair-end eclip data. For single-end eCLIP data, use UMI-tools for deduplication.

barcodecollapsepe.py can be retrieved from yeo's lab Github.
The script is under the bin directory.

Anaconda

Anaconda environment is a pre-installed tool in Bridges-2. It is a data science platform which includes Python and R. Multiple versions of Anaconda are available on Bridges-2. For this pipeline, Anaconda3 is installed for deepTools and PEAKachu packages. To add the environment on Bridges-2, refer to the Adding to an environment section: Anaconda | PSC. Then install the following packages.

deepTools

deepTools is a suite of python tools particularly developed for the efficient analysis of high-throughput sequencing data, such as ChIP-seq, RNA-seq or MNase-seq. Installation is through the Bioconda channel in the Anaconda environment. We use *plotFingerprint*, *multiBamSummary*, and *plotCorrelation* tools to perform the second quality control for the mapped reads.

For installation details, refer to: Installation — deepTools
For more information, refer to: deepTools

PEAKachu

PEAKachu is a peak calling tool for CLIP-seq data. Installation is through the Bioconda channel in the Anaconda environment. It takes input in BAM format and identifies regions of statistically significant read enrichment. PEAKachu uses signal and control libraries to detect binding sites. It implements two peak calling approaches.

For installation details, refer to: Peakachu
For usages, refer to: PEAKachu

MEME Suite

MEME Suite is a collection of pre-installed tools in Bridges-2, which is used for the discovery and analysis of sequence motifs. We use the MEME-ChIP tool to discover novel motifs. The Position-Specific Weight Matrix (PSWM) is the core of this algorithm, which is a probability matrix of bases at different positions of a sequence. A motif can be considered as a pattern among multiple sequences, so that it is possible to distinguish between motif sequences and background sequences based on their different position-specific weight. Normally, the parameters of this probabilistic model are estimated via estimation-maximization algorithm, where in the estimation step we randomly assign parameters and in the maximization step we update parameters to maximize the expected likelihood.

For usage on Bridges-2, refer to: MEME-Suite | PSC
For usages of MEME-ChIP, refer to: MEME-ChIP

Cthreepo

Cthreepo is a python script that helps convert reads whose chromosome name in NCBI format to the chromosome name used by the

UCSC system. It supports various input file formats, such as GFF, GTF, bed and so on.

To install Cthreepo, refer to: Cthreepo-Github

RNA Centric Annotation System

RNA Centric Annotation System (RCAS) is a R package and it conducts functional analysis on transcriptome-wide regions. It provides various analysis, including annotation summaries, GO term enrichment analysis, gene set enrichment analysis and motif discovery.

To learn more details, refer to the original paper: RCAS
For installation and usage instructions, refer to: RCAS user guideline

Pipeline Instructions

Required Input Files

Part 1: Motif Discovery (bridges2 supported)

The reference data file can be downloaded by the users from the public database: Human Genome Resources at NCBI.
The experimental data can be generated either by the users or downloaded from the public ENCODE database. Link to the website that contains the files used in our pipeline: Experiment summary for ENCSR841EQA.
The control data for HepG2 cell lines in ENCODE can be retrieved from Mock Input for HepG2.

1. Reference genome fasta file: serves as reference genome file for STAR Aligner.
2. Reference genome annotation GFF or GTF file: genome annotation file, which can be in either GFF or GTF format.

3. Two pair-end eCLIP experiment fasta files: experimental data to build our own transcripts. These include experiment_read1.fastq and experiment_read2.fastq.
4. Two pair-end eCLIP control fasta files: HepG2 control data to build our own transcripts. These include control_read1.fastq and control_read2.fastq.
5. barcodecollapsepe.py python script: used for deduplication of paired-end eCLIP data after alignment.
6. (The 8th) hg38_acc_sizes.txt: it has the size of each chromosome (labeled by their accession ID)

NOTICE: Due to different versions of the python you have installed on your computer, you may need to manually replace “itertools.izip” statement in barcodecollapsepe.py you download from yeo’s lab github with “zip”. In newer python3. The itertools.izip is no longer used and replaced by the built-in “zip.”

Part 2: Additional downstream analysis (locally supported)

Downstream_analysis.r R script for sequence annotation and GO analysis.

NOTICE: You should have R installed on your computer. Test it with “R —version” in your terminal.

Usage

Part 1: MOTIF DISCOVERY

As a bash script, it can be simpy run with one command in the terminal:

sbatch pipeline_name.sh ***$1 $2 $3 $4 $5 $6 $7 $8***
(on bridges)

$1: the absolute path to the reference genome fasta file
$2: the absolute path to the reference genome annotation GFF/GTF file
$3: the absolute path to the experiment pair-end eclip read_1 fastq file
$4: the absolute path to the experiment pair-end eclip read_2 fastq file
$5: the absolute path to the control pair-end eclip read_1 fastq file
$6: the absolute path to the control pair-end eclip read_2 fastq file

$7: the absolute path to the barcodecollapsepe.py python file
$8: the absolute path to the hg38_acc_sizes.txt

Part 2: DOWNSTREAM ANALYSIS (Additional)

The downstream analysis cannot be run on bridges, but it can be done locally as an additional analysis based on the discovered peaks. There are two scripts used for this part, including one bash script and one R script. To run this, make sure you are under your start directory and give the following command in your terminal:

*bash downstream_analysis.sh **$1 $2***

$1: the absolute path to the reference genome annotation GFF/GTF file
$2: the absolute path to the downstream_analysis.r R script (in case you change its directory)

Pipeline Output

Directories Structure

If the motif discovery pipeline works well, the output directories will be:

your start directory/
step1_first_quality_control/
step2_adapter_trimming/
    exptrimmed_round1_1.fastq
    exptrimmed_round1_2.fastq
    exptrimmed_round2_1.fastq
    exptrimmed_round2_2.fastq
    contrimmed_round1_1.fastq
    contrimmed_round1_2.fastq
    contrimmed_round2_1.fastq
    contrimmed_round2_2.fastq
step3_genome_indice/
step4_alignment/
    experiment/
        expAligned.sortedByCoord.out.bam
        exp_mapped.bam
        expsorted.bam
        expnamesorted.bam
        expnamesorted.bam.bai (from expsorted.bam)
    control/
        conAligned.sortedByCoord.out.bam
        con_mapped.bam
        consorted.bam
        connamesorted.bam

```
        connamesorted.bam.bai (from consorted.bam)
step5_deduplication/
        expnamesorteddup.bam
        expdupsorted.bam
        expdupsorted.bam.bai
        connamesorteddup.bam
        condupsorted.bam
        condupsorted.bam.bai
step6_second_quality_control/
step7_coverage_analysis/
          Fingerprint.png
step8_correlation_analysis/
          multiBamSummary/
                multiBamSummaryresults.npz
                correlation.png
step9_peak_calling/
          peak_annotations/
          peak_tables/
          plots/
          Initial_peaks.csv
step10_meme_chip/
          memechip_out/
                meme-chip.html
                motif_alignment.txt
                Summary.tsv
```

If you are planning to conduct the additional downstream analysis, there will be one more folder generated:

```
step11_downstream_analysis/
        annotation.png
        GO.png
```

Overall Standard Output and Error Information

There might be some errors returned by the tools due to inappropriate usage or environment setup. Our pipeline separately generates flow.sh.err file to keep track of error information and facilitates user debugging. There is another system file named flow.sh.out, which records the standard output of the pipeline, including the time consumption and processing information. Note that the information in the flow.sh.err is not necessarily an “error,” but it could possibly be feedback messages from the packages.

FastQC

FastQC can compare the quality reports given by FastQC before and after the deduplication of PCR products during the data pre-processing period to check whether the deduplication is effective.

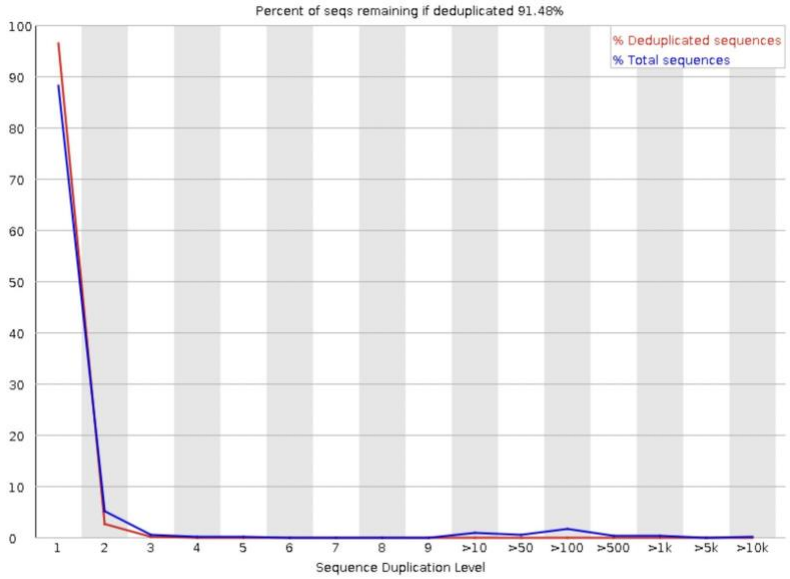


Figure 1: Duplication analysis from FastQC.

Cutadapt

Cutadapt trims off the adaptor sequences attached to the reads during PCR. As we are using pair-end eclip data here, the adaptor trimming procedure is performed for two rounds, which respectively processes the 3' end and 5' end of the sequences. For each type of data (experiment or control), the Cutadapt will generate four fastq files, where the first two fastq files after round 1 trimming will be directly used as the input for the second trimming process.

STAR Aligner

STAR Aligner will first generate bam files which would be in the step4_ folder for aligning the eCLIP-seq reads to the reference genome. Samtools is then used to make a bam file. Then Star Aligner will index the reference genome using the newly assembled experimental bam file. To do this, we will create a directory to hold these index files. Then STAR Aligner

aligns the eCLIP-seq reads to the indexed experimental genome, creating a sam file. The sam file is then converted to a bam file and sorted using SAMtools.

Samtools

In the pipeline, Samtools is mainly used to sort and index the bam file to make it a valid input for barcodecollapsepe.py and PEAKachu in the. In the pre-deduplication step, first of all, it processes the bad mapped reads by the option *view -b -f 8*. Sometimes one read from the pair-end data is not mapped to the reference genome, which may cause a problem to the following steps and should be filtered out. Next, the *name sort* option in Samtools is used to generate a name-sorted alignment bam file. In the pre-peak-calling step, Samtools is used to sort and index the aligned bam file from STAR alignment as a preparation of PEAKachu peak calling, where index file (.bai) is stored under the same directory of the bam file for reference.

barcodecollapsepe.py

The generated bam file from the adaptor trimming step needs to be deduplicated. The barcodecollapsepe.py takes the name-sorted bam file as input and performs deduplication. Notice that due to the version issue, it sometimes may return an error, saying that no index file is found, which you can ignore as the script does not really need an index file to run. A deduplicated bam file will be generated and the generated deduplicated file will be sorted and indexed again as required by the next peak calling step.

deepTools

We check the quality of our mapped reads and see if our samples are correlated or not. As some of the file sizes are fairly large, it would be wise to check if some samples may encompass major quality problems.

plotFingerprint

plotFingerprint randomly samples genome regions of a specified length and sums the per-base coverage that overlap with those regions. It shows us how good the CLIP Signal compared to the control signal is. A negative control often has the sharp slope at the end as a CLIP experiment but often depicts a straighter, diagonal line in the beginning like the input control.

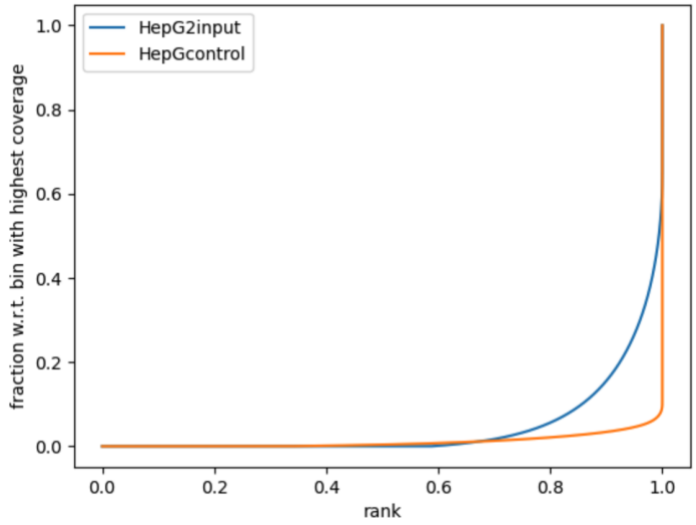


Figure 2: Graph of IP strength from plotFingerprint.

plotCorrelation

plotCorrelation is similar to plotFingerprint which also randomly samples genome regions of a specified length and sums the per-base coverage that overlap with those regions. It shows the correlation between these bins for each pair of samples. This is necessary as our input control and our CLIP experiment might be strongly correlated, which means that the potential protein binding regions are not truly enriched when compared to our control. An ideal plot would show two disparate clusters, one for the biological replicates of the CLIP-Seq experiment and one for the replicates of the control.

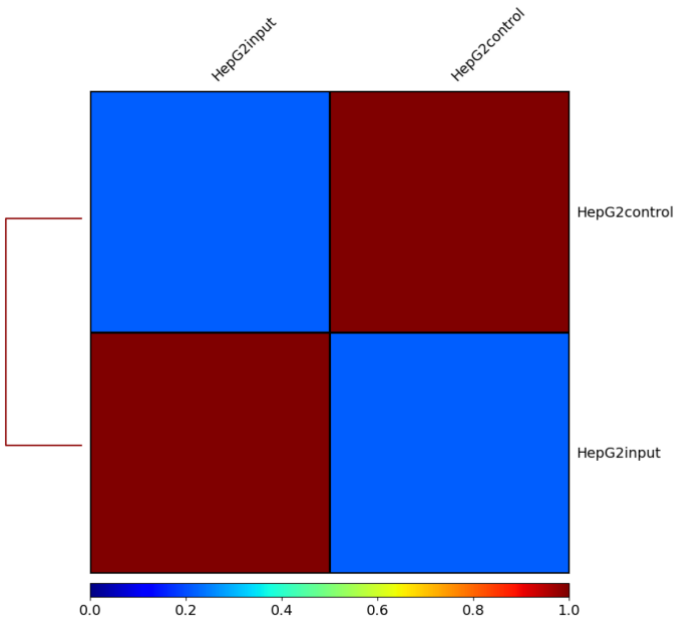


Figure 3: Heatmap of correlation matrix generated by plotCorrelation.

PEAKachu

PEAKachu is able to incorporate control data in contrast to other peak callers like Piranha, thus allowing us to find binding regions that are significantly enriched in comparison to our control input data. PEAKachu is well suited for an experimental set-up with more than 2 replicates for the CLIP experiment and more than 2 replicates for the control experiment because it uses DESeq2. It will output all of the peaks in a csv file for MEME-ChIP to find motifs. The peaks are indexed by their accession numbers with the start and the end locations of the peak.

Table 1. Peak table generated by PEAKachu.

replicon	peak_start	peak_end	peak_strand	resultAligned.sortedByCoord.out	control.out	base_means	fold_change
NC_000001.11	21455	21499	-	11.3137085	28.28427125	19.79898987	0.4
NC_000001.11	21724	21767	-	8.131727984	8.485281374	8.308504679	0.9583333333
NC_000001.11	21801	21864	-	19.09188309	11.3137085	15.2027958	1.6875
NC_000001.11	22652	22725	-	21.56675683	5.656854249	13.61180554	3.8125
NC_000001.11	22408	22484	-	32.52691193	22.627417	27.57716447	1.4375
NC_000001.11	22969	23020	-	8.131727984	5.656854249	6.894291117	1.4375
NC_000001.11	22247	22298	-	10.25304833	8.485281374	9.369164851	1.208333333

MEME Suite

MEME-ChIP tool is used to discover novel motifs in the peak csv file generated by PEAKachu. MEME-ChIP can create a GFF file for viewing each motif's predicted sites in a genome browser.

Figure 4: Motifs found by MEME-ChIP.

Downstream Analysis

Downstream analysis uses an R script sourced from RNA Centric Annotation System, which is modified to be suitable for usage for continuation of this pipeline. It will annotate the peaks with different sequence structures and give a GO analysis of the RNA.

	p_value	term_name	source
8	0	negative regulation of cellular process	GO:BP
9	0	regulation of molecular function	GO:BP
10	0	developmental process	GO:BP
11	0	anatomical structure development	GO:BP
12	0	cellular macromolecule metabolic process	GO:BP
13	0	regulation of catalytic activity	GO:BP
14	0	negative regulation of cellular metabolic process	GO:BP
15	0	organonitrogen compound metabolic process	GO:BP
16	0	positive regulation of biological process	GO:BP
17	0	negative regulation of nitrogen compound metabolic process	GO:BP

Figure 5: GO terms matched for the RNA sequence.

Pitfalls and Limitations

The current pipeline is built upon a lot of other external packages which might cause problems in version compatibility. If one of the packages updated or made changes, our pipeline risks not running smoothly as expected.
The pipeline requires a large amount of computational power, and so it is designed to be run on PSC. However, because PSC has a different system environment than many other computation systems, the current pipeline might not be compatible with all computing environments.