



ChromAcS – User Manual

Version: 1.0.0

Authors: ChromAcS Development Team

Project Link: <https://github.com/epigen-bioinfolab/CHROMACS>

Introduction

ChromAcS (**Chromatin Accessibility Analysis Suite**) is a user-friendly, GUI-based, end-to-end pipeline for analyzing paired-end ATAC-seq data. ChromAcS integrates best-practice tools into a single cohesive workflow to perform automated quality control, adapter trimming, read alignment, coverage track generation, peak calling, genomic peak annotation, and differential accessibility analysis. It further incorporates motif enrichment and regulatory network analysis, motif footprinting, peak overlap comparison, and integration with gene expression datasets, enabling versatile downstream multi-omic exploration. The pipeline supports reference genome handling across multiple species, peak annotation, and multiple modes of differential accessibility analysis, including both replicate-based and no-replicate scenarios.

ChromAcS is a GUI-based ATAC-seq analysis platform with multi-species support, enabling real-time progress monitoring, flexible re-execution of downstream steps without restarting the workflow, and easy deployment in a Linux environment using our automated setup script for installing all required dependencies.

Installation Guide

ChromAcS is primarily developed and tested for Linux-based platforms (e.g., Ubuntu, Fedora, Debian), which are recommended for full compatibility and optimal performance. While Linux is the officially supported environment, ChromAcS may also function on other Unix-like systems such as macOS, provided the necessary dependencies are met. Use on Windows is not officially supported, but advanced users may attempt installation via WSL (Windows Subsystem for Linux) or a virtual machine running a Linux distribution.

To ensure reproducibility and to prevent conflicts with other software, ChromAcS is designed to operate within an isolated Conda environment. This simplifies installation and management of the required packages and dependencies, without clashing with other tools.

The installation process is straightforward and involves the following steps:

Step 1: (Clone ChromAcS repository or download as a ZIP):

To clone the repository and navigate to it, just paste the following after opening the terminal-

```
git clone https://github.com/epigen-bioinfolab/CHROMACS.git
cd CHROMACS
```

Alternatively, you can download the zip file-

1. Go to <https://github.com/epigen-bioinfolab/CHROMACS>
2. Click the green <> **Code** button → Download ZIP
3. Extract the ZIP file. The folder will be named CHROMACS-main. Configure to the CHROMACS-main-

```
cd CHROMACS-main
```

Step 2: Create and activate the environment :

If you have followed the above steps, you should find yourself inside the CHROMACS (or, CHROMACS-main) directory on the terminal (be certain that you are indeed inside such a directory; if not, you can manually click to search and go inside the directory). In your terminal, paste the following-

```
conda env create -f environment.yml
conda activate chromacs
```

Note (a prerequisite for Step 2):

Step 2 might take several minutes to set up the conda environment. However, please ensure that you have Conda installed on your system. For example, you can install Miniconda as follows-

```
# Download the Miniconda installer (64-bit)
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

```
# Run the installer
bash Miniconda3-latest-Linux-x86_64.sh
```

```
# Follow the prompts and restart your terminal (or run 'source ~/.bashrc')
```

Step 3: Install the tool:

Paste this on your terminal (ensure that you are indeed inside the CHROMACS environment setup in conda)-

```
pip install . # DO NOT FORGET THE DOT (.) at the end "pip install ."
```

Step 4: Launch the tool:

Just type the following in the terminal to launch the base tool

```
Chromacs
```

Step 4a (Add-on): Launch the ChromAcS-AddOn: Additional Analysis Toolkit :

```
chromacs-addon
```

Extra guides:-

How to relaunch the tool once you close it?

After installing and running the tool, you may shut the tool. Now, to relaunch it, the steps are even easier, just open your terminal anywhere (you do not need to be inside the CHROMACS directory now) and activate conda (type the following in your terminal)-

```
conda activate chromacs  
chromacs
```

How to uninstall the tool?

If you wish to uninstall ChromAcS, you need to navigate to the CHROMACS directory, then type in the following in your terminal-

```
pip uninstall chromacs -y
```

Remove the associated Conda environment-

```
conda env remove -n chromacs
```

Navigating inside ChromAcS

ChromAcS provides a simple and straightforward navigation for the users, so that they can easily set up the different parameters accordingly. Here, we provide a step-by-step guide on how we used ChromAcS to process ATAC-seq data from two related NCBI GEO BioProjects- GSE100750 (BioProject: PRJNA392915) and GSE73874 (BioProject: PRJNA301678), both published by Guler *et al* (<https://pubmed.ncbi.nlm.nih.gov/28781121>). These datasets profile the chromatin accessibility in PC9 lung cancer cell lines and their drug-tolerant persister (DTP) derivatives under various treatment conditions. We select two experimental groups to simulate a two-condition ATAC-seq differential accessibility study, where condition 1 consists of the

untreated parental PC9 cells, and condition 2 consists of DTPs (these are drug-tolerant subpopulations of cells that survived erlotinib, an EGFR inhibitor, treatment).

Conditions	Sample IDs	Control Sample IDs
Condition 1 (untreated)	SRR5800657 (GSM2692559)	SRR2927084 (GSM1904729)
	SRR5800658 (GSM2692560)	
	SRR5800659 (GSM2692561)	
Condition 2 (treated/DTPs)	SRR5800660 (GSM2692562)	SRR2927085 (GSM1904730)
	SRR5800661 (GSM2692563)	
	SRR5800662 (GSM2692564)	

The three samples in condition 1 and condition 2 are biological replicates of parental PC9 cells and DTPs, respectively.

Step 1: Setting the base output directory and the raw data directory

You need to put the raw paired-end ATAC-seq data in a single directory and make a new directory where all the outputs generated by ChromAcS will be saved. Then inside ChromAcS, you need to browse inside those-

How to browse it right

Here, you must remain careful that while browsing the required directories, you must navigate inside the given directory. For example, here our raw data is saved inside the

“home/prince/atac_seq_project/raw_data”, so while browsing, we must go inside the raw_data itself (users may often mistakenly just move inside the atac_seq_project directory and see the raw_data, but not click inside it. But you MUST click inside the directory itself).

Thread setup

Another parameter to set up here is the number of threads that would be assigned to several of the dependencies inside ChromAcS. We would recommend keeping it to the default of 8 if you have 16 GB of memory (RAM). If you have lower RAM, you may decrease the thread count.

Processing all the samples?

Note that, if you keep the “Select specific samples” UNCHECKED, all the paired-end files inside the “Raw Data Directory” will be processed in the downstream step as you can check in the next step (Step 2)-

Step 1: Data Setup **Step 2: Quality Control** Step 3: Reference Selection Step 4: Peak Caller Selection Step 5: Analysis

Auto-Detected Sample Names:

SRR2927084
SRR2927085
SRR5800657
SRR5800658
SRR5800659
SRR5800660

Please HOVER on this box and SCROLL to see more samples (if available)

Samples auto-extracted from filenames.
Format: SRRXXXXX (ignores _1/_2 suffixes).

A list of allowed sample name types detected by ChromAcS-
sample_1.fastq.gz / fq.gz; sample_2.fastq.gz / fq.gz
sample_R1.fastq.gz / fq.gz; sample_R2.fastq.gz / fq.gz
sample_S1_L002_R1_001.fastq.gz / fq.gz; sample_S1_L002_R2_001.fastq.gz / fq.gz

☐ **Skip Trimming (Use Raw Data for Alignment)**

Checking this will skip the trimming done by Trim Galore.
Raw data will be used directly for alignment.
Please check the FastQC and MultiQC reports if you are unsure of this step.

Back **Save & Next**

As you can see, the auto-detected sample names detected all the samples (here, you see only six of the sample names; scrolling down in that box will show the other two samples as well.) We had a total of 8 samples, that is, 16 reads. Now, there are some prerequisites for the file names of samples, and we have tried to add most of the naming conventions typically used.

Which names will be valid for a successful downstream processing in ChromAcS?

sample_1.fastq.gz / fq.gz; sample_2.fastq.gz / fq.gz

sample_R1.fastq.gz / fq.gz; sample_R2.fastq.gz / fq.gz

sample_S1_L002_R1_001.fastq.gz / fq.gz; sample_S1_L002_R2_001.fastq.gz / fq.gz

Here, the first part is the forward reads and the second part (after a semi-colon) is the reverse reads.

Processing Selective samples?

This can be done simply by CHECKING the “Select specific samples” in step 1-

Step 1: Data Setup | Step 2: Quality Control | Step 3: Reference Selection | Step 4: Peak Caller Selection | Step 5: Analysis

Base Output Directory: Browse Create

*[Assign the Base Output Directory where all the corresponding results will be saved.]
"Create" allows a new output directory; "Browse" to use an existing directory already created*

Raw Data Directory: Browse

[If 'Select specific samples' is unchecked, ALL FASTQ files in the RAW DATA DIRECTORY will be processed.]

☒ Select specific samples

SRR2927084_1.fastq.gz
SRR2927084_2.fastq.gz
SRR2927085_1.fastq.gz
SRR2927085_2.fastq.gz
SRR5800657_1.fastq.gz
SRR5800657_2.fastq.gz

Number of Threads (Default- 8):

Save & Next

Once you CHECK the “Select specific samples”, a new text widget appears, showing the samples. You can click on the required samples to select. However, be certain to include both reads of the paired-end samples. Once you are done selecting the specific samples, you can verify those again in step 2-

Step 1: Data Setup | **Step 2: Quality Control** | Step 3: Reference Selection | Step 4: Peak Caller Selection | Step 5: Analysis

Auto-Detected Sample Names:

SRR2927084
SRR5800657
SRR5800660

Please HOVER on this box and SCROLL to see more samples (if available)

*Samples auto-extracted from filenames.
Format: SRRXXXXX (ignores _1/_2 suffixes).*

*A list of allowed sample name types detected by ChromAcS-
sample_1.fastq.gz / fq.gz; sample_2.fastq.gz / fq.gz
sample_R1.fastq.gz / fq.gz; sample_R2.fastq.gz / fq.gz
sample_S1_L002_R1_001.fastq.gz/ fq.gz; sample_S1_L002_R2_001.fastq.gz/ fq.gz*

☐ **Skip Trimming (Use Raw Data for Alignment)**

*Checking this will skip the trimming done by Trim Galore.
Raw data will be used directly for alignment.
Please check the FastQC and MultiQC reports if you are unsure of this step.*

Back Save & Next

Step 2: Verifying the selected samples and choosing trimming

This step is pretty straightforward, as you verify here the samples detected by ChromAcS inside your assigned Raw_data_directory. You can also decide in this step whether you want to allow ChromAcS to perform the trimming operation or not. ChromAcS pipeline uses Trim Galore for trimming the reads.

Step 1: Data Setup | **Step 2: Quality Control** | Step 3: Reference Selection | Step 4: Peak Caller Selection | Step 5: Analysis

Auto-Detected Sample Names:

- SRR2927084
- SRR2927085
- SRR5800657
- SRR5800658
- SRR5800659
- SRR5800660

Please HOVER on this box and SCROLL to see more samples (if available)

Samples auto-extracted from filenames.
Format: SRRXXXXX (ignores _1/_2 suffixes).

A list of allowed sample name types detected by ChromAcS-
sample_1.fastq.gz / fq.gz; sample_2.fastq.gz / fq.gz
sample_R1.fastq.gz / fq.gz; sample_R2.fastq.gz / fq.gz
sample_S1_L002_R1_001.fastq.gz / fq.gz; sample_S1_L002_R2_001.fastq.gz / fq.gz

☐ **Skip Trimming (Use Raw Data for Alignment)**

Checking this will skip the trimming done by Trim Galore.
Raw data will be used directly for alignment.
Please check the FastQC and MultiQC reports if you are unsure of this step.

Back **Save & Next**

What happens if you skip trimming?

If you CHECK “Skip Trimming (Use Raw Data for Alignment)”, then ChromAcS will align the raw Fastq reads in the subsequent steps. This will skip the trimming steps by Trim Galore.

Can you use your own trimmed reads?

Yes, you can use your own trimmed reads, that is, if you prefer to trim your data with a separate trim tool (like trimmomatic, BBDDuk, etc.), then you can rename the output with the naming conventions suggested in the tool. This you just simply use this reads in the raw data directory, and MUST CHECK “Skip Trimming (Use Raw Data for Alignment)”. In that way, you would be using your own trimmed data, if you are unsure to use our default Trim Galore trimming.

Step 3: Selecting the Desired Organism/ Reference Genome

In this step, you need to assign the organism you are studying. Currently, ChromAcS support the following list of organisms-

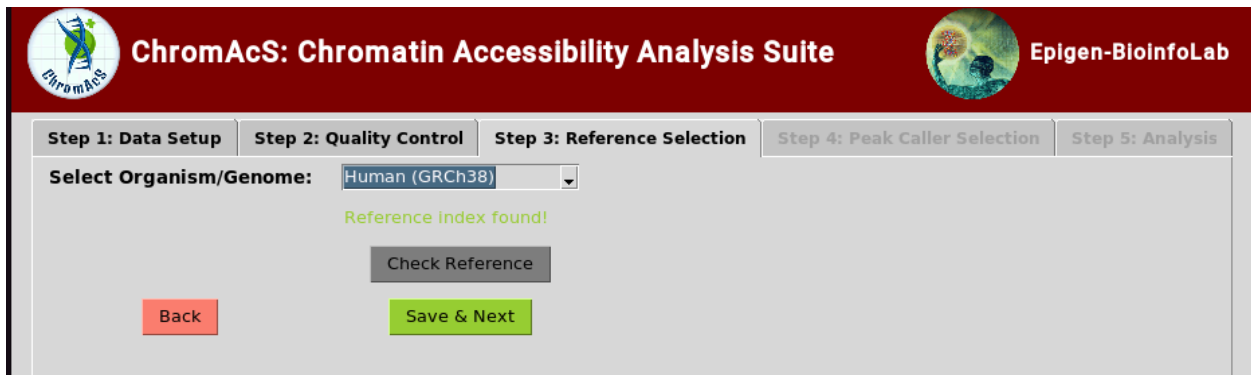
Table 1. List of organisms supported

Common Name	Scientific Name	Genome Version
Human	<i>Homo sapiens</i>	GRCh38
Mouse	<i>Mus musculus</i>	GRCm39
Rat	<i>Rattus norvegicus</i>	mRatBN7.2
Cow	<i>Bos taurus</i>	ARS-UCD1.3
Pig	<i>Sus scrofa</i>	Sscrofa11.1
Chicken	<i>Gallus gallus</i>	bGalGal1.mat.broiler.GRCg7b
Chimpanzee	<i>Pan troglodytes</i>	Pan_tro_3.0
Dog	<i>Canis lupus familiaris</i>	ROS_Cfam_1.0
Goat	<i>Capra hircus</i>	ARS1
Rabbit	<i>Oryctolagus cuniculus</i>	OryCun2.0
Gorilla	<i>Gorilla gorilla</i>	gorGor4
Rhesus Macaque	<i>Macaca mulatta</i>	Mmul_10
Zebrafish	<i>Danio rerio</i>	GRCz11
Frog	<i>Xenopus tropicalis</i>	UCB_Xtro_10.0
Indian Cobra	<i>Naja naja</i>	Nana_v5
Atlantic Salmon	<i>Salmo salar</i>	Ssal_v3.1
C. elegans	<i>Caenorhabditis elegans</i>	WBcel235
Drosophila	<i>Drosophila melanogaster</i>	BDGP6

This step is crucial as it determines which reference genome to build and index, which GTF file to download and process in the later steps, and also very crucial for the annotation steps.

Does ChromAcS build and index the reference genome on every run?

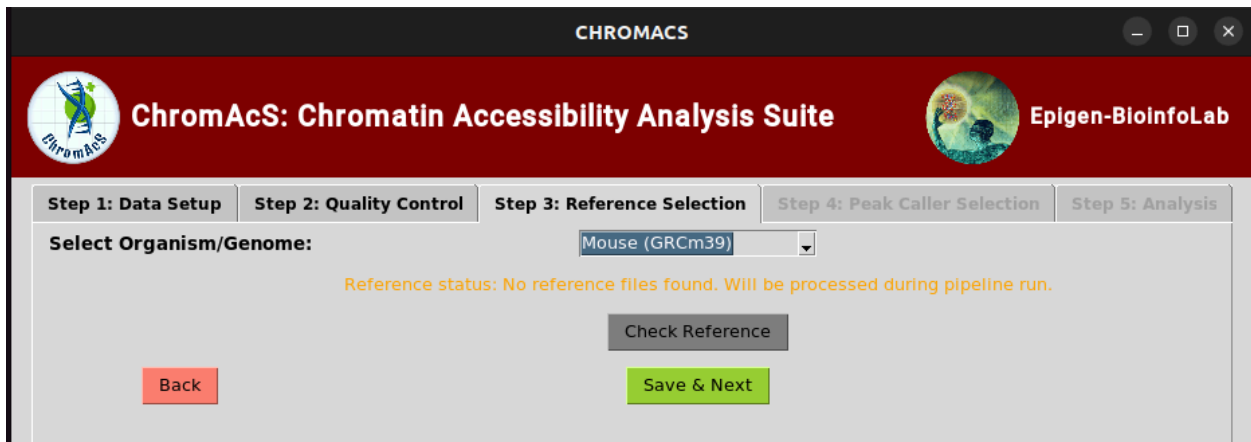
No, it only does it for the first run for a given reference genome (thus, it takes some time on the first run for a given reference genome, around 90-120 minutes extra depending on thread setup and internet connection speed). But once you have already run ChromAcS for a given reference genome, it saves the Bowtie2 built indices in a separate directory within the “chromacs” conda environment, along with the corresponding GTF file, the reference TSS file, the TxDB object, and the org.db file (used during annotation later). This makes the next runs smoother and saves a significant amount of time. You can check whether the reference genome has already been built or not by clicking “Check Reference” in step 3.



The screenshot shows the ChromAcS web interface. The top header is red with the ChromAcS logo on the left and the Epigen-BioInfoLab logo on the right. Below the header is a navigation bar with five tabs: Step 1: Data Setup, Step 2: Quality Control, Step 3: Reference Selection (active), Step 4: Peak Caller Selection, and Step 5: Analysis. The main content area is for Step 3. It has a label "Select Organism/Genome:" followed by a dropdown menu showing "Human (GRCh38)". Below the dropdown, the text "Reference index found!" is displayed in green. There are three buttons: a red "Back" button, a grey "Check Reference" button, and a green "Save & Next" button.

As you can see here, the message says “Reference index found!”, suggesting that the user has already run the tool once for Human (GRCh38). And thus, during the current run, the building will be avoided.

If it was not run, and for the first run, a following message will be shown on clicking the “Check Reference”-



The screenshot shows the ChromAcS web interface. The top header is red with the ChromAcS logo on the left and the Epigen-BioInfoLab logo on the right. Below the header is a navigation bar with five tabs: Step 1: Data Setup, Step 2: Quality Control, Step 3: Reference Selection (active), Step 4: Peak Caller Selection, and Step 5: Analysis. The main content area is for Step 3. It has a label "Select Organism/Genome:" followed by a dropdown menu showing "Mouse (GRCm39)". Below the dropdown, the text "Reference status: No reference files found. Will be processed during pipeline run." is displayed in orange. There are three buttons: a red "Back" button, a grey "Check Reference" button, and a green "Save & Next" button.

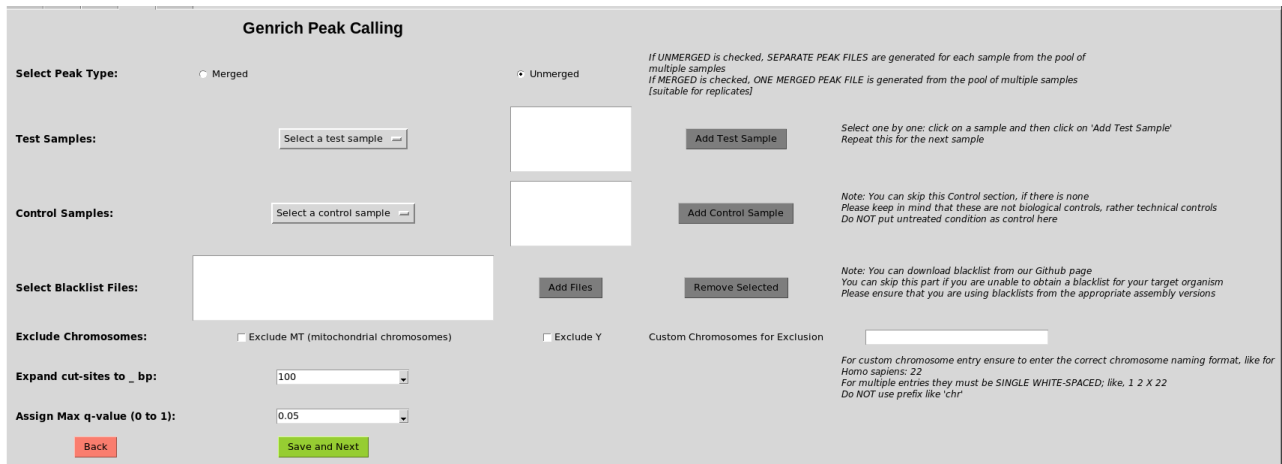
Step 4: Selection of Peak Caller and Adjusting the Parameters

This is one of the most crucial steps, where the user needs to select a desired peak caller from two options: a) Genrich or b) MACS3.



The screenshot shows the CHROMACS web interface. The top navigation bar includes the CHROMACS logo, the title "ChromAcS: Chromatin Accessibility Analysis Suite", and the Epigen-BioInfoLab logo. Below the navigation bar, there are five tabs: "Step 1: Data Setup", "Step 2: Quality Control", "Step 3: Reference Selection", "Step 4: Peak Caller Selection" (which is active), and "Step 5: Analysis". The main content area is titled "Peak Calling" and contains the "Select Peak Caller:" section. It has two radio buttons: "Genrich" (selected) and "MACS3". At the bottom of this section are two buttons: "Back" (red) and "Save & Next" (green).

Once you have selected a peak caller, it takes you to the parameter setup, where you can adjust several parameters. The parameters are more or less similar for both peak callers-



The screenshot shows the "Genrich Peak Calling" parameter setup form. It includes several sections: "Select Peak Type:" with radio buttons for "Merged" and "Unmerged" (selected); "Test Samples:" with a "Select a test sample" dropdown and an "Add Test Sample" button; "Control Samples:" with a "Select a control sample" dropdown and an "Add Control Sample" button; "Select Blacklist Files:" with a text input field, "Add Files", and "Remove Selected" buttons; "Exclude Chromosomes:" with checkboxes for "Exclude MT (mitochondrial chromosomes)" and "Exclude Y", and a "Custom Chromosomes for Exclusion" text input; "Expand cut-sites to _bp:" with a dropdown set to "100"; and "Assign Max q-value (0 to 1):" with a dropdown set to "0.05". There are "Back" (red) and "Save and Next" (green) buttons at the bottom. A note on the right side explains the difference between UNMERGED and MERGED peak files and provides instructions for the custom chromosome entry.

(Genrich parameter setups)

MACS3 Peak Calling

Test Samples: Select one by one: click on a sample and then click on 'Add Test Sample'
Repeat this for the next sample; you can ADD MULTIPLE SAMPLES at a time

Control Samples: Note: You can skip this Control section, if there is none
Please keep in mind that these are not biological controls, rather technical controls
Do NOT put untreated condition as control here

Select Blacklist Files: Note: You can download blacklist from our Github page
You can skip this part if you are unable to obtain a blacklist for your target organism
Please ensure that you are using blacklists from the appropriate assembly versions

Exclude Chromosomes: ☐ Exclude MT (Mitochondrial chromosomes) ☐ Exclude Y Custom Chromosomes for Exclusion:

Assign Max q-value (0 to 1):

Genome Size (auto-filled):

For custom chromosome entry ensure to enter the correct chromosome naming format, like for Homo sapiens: 22
For multiple entries they must be SINGLE WHITE-SPACED; like, 1 2 X 22
Do NOT use prefix like 'chr'

(MACS3 parameter setups)

How to set up the test samples?

You need to add the test sample one by one; that is, click on the “Select a test sample” and the test samples will appear. From there, you need to click on one test sample and then click on the “Add Test Sample”. You need to repeat this for multiple test samples-

Test Samples:

Control Samples:

Select Blacklist Files:

SRR2927084
SRR2927085
SRR5800657
SRR5800658
SRR5800659
SRR5800660
SRR5800661
SRR5800662

(When you click on the test sample, a drop-down menu appears with all the samples, from which you click on one sample at a time.)

Test Samples:

Control Samples:

Select Blacklist Files:

(Once you have clicked on a desired sample, it appears on the test sample, but it has not been added to the list to be processed yet for peak calling.)

The screenshot shows a web interface for sample selection. It has three main sections: 'Test Samples:', 'Control Samples:', and 'Select Blacklist Files:'. In the 'Test Samples:' section, a dropdown menu shows 'SRR5800657' and an 'Add Test Sample' button is to its right. In the 'Control Samples:' section, there is a dropdown menu with the text 'Select a control sample' and an 'Add Control Sample' button to its right. In the 'Select Blacklist Files:' section, there is a large empty text box, an 'Add Files' button, and a 'Remove Selected' button.

(When you click on the “Add Test Sample”, the desired sample will appear in the list box to be used during peak calling.)

You need to repeat this step for all the test samples you need to process, and for both conditions. For instance, in our case study, we added all six test samples for both conditions - treated and untreated in these test samples.

This screenshot shows the same interface as the previous one, but the 'Test Samples:' dropdown menu is now open, displaying a list of six sample IDs: SRR5800657, SRR5800658, SRR5800659, SRR5800660, and SRR5800661. The 'Add Test Sample' button remains to the right of the list. The 'Control Samples:' and 'Select Blacklist Files:' sections are unchanged.

(All six test samples have been added; here you see only five, another one can be seen by scrolling down by hovering the mouse over that particular list box)

How to set up the control samples?

The user needs to keep in mind that the control samples here mean the technical controls used by the peak callers to clear some of the background signals during peak calling. So, they are not the conventional controls in the biological sense, like we should not assign our “untreated” condition samples as control samples here. Rather, our control samples were SRR2927084 for the untreated test samples, and SRR2927085 for the treated/DTP samples.

In such designs where we have two different technical controls for the two conditions, the peak calling parameters need to be adjusted accordingly-

Test Samples:	<input type="text" value="SRR5800659"/>	<input type="text" value="SRR5800657"/> <input type="text" value="SRR5800658"/> <input type="text" value="SRR5800659"/>	<input type="button" value="Add Test Sample"/>
Control Samples:	<input type="text" value="SRR2927084"/>	<input type="text" value="SRR2927084"/>	<input type="button" value="Add Control Sample"/>

(Setting up the untreated condition test samples and the required technical control. With this design, we need to run the entire pipeline once. Then we repeat the process for the treated condition test samples.

Test Samples:	<input type="text" value="SRR5800662"/>	<input type="text" value="SRR5800660"/> <input type="text" value="SRR5800661"/> <input type="text" value="SRR5800662"/>	<input type="button" value="Add Test Sample"/>
Control Samples:	<input type="text" value="SRR2927085"/>	<input type="text" value="SRR2927085"/>	<input type="button" value="Add Control Sample"/>

(setting up the treated condition test samples, and the required technical control. So we can not run ChromAcS with the different technical controls for both conditions at a time; rather, in such designs, we need to run the tool twice, selectively for both conditions.

What if you do not have the technical controls in your study design?

You can run the tool regardless of technical controls; it is optional. We would suggest running without the technical controls, as the technical controls may limit/ exclude many of the relevant biological peaks due to stringency.

Setting up the miscellaneous parameters

The rest of the parameters can be set up by the user according to their preference, for example, setting up the q-value, the d-value (exclusive in Genrich), Blacklist regions, and the Exclusion regions.

For Blacklist Files, we would suggest only those BED files where the chromosome naming is in Ensembl formats, that is, like 1, 2, 3, 12, 21, X, Y (not, chr1, chr2, chrX). If you do not find relevant blacklist files, it is fine to keep this empty.

For the other exclusion regions, you can select the default MT exclusion, Y chromosome exclusion (we added these two exclusively since in most studies it is relevant to exclude these two). However, you can type in a custom set of chromosomes in the text box for exclusion, which must be space-separated, like 1 2 22 12 (not like 1, 2, 22, 12 or etc.) This step is also optional.

For Genrich, you can mention the exact bp of expanding cut sites, that is, the d-value. You can select any relevant numbers from the drop-down menu, or you can even type in your own desired value. The default value is 100.

You can assign the q-value in a similar manner. The default value is 0.05.

An important consideration for Blacklist files

Since the blacklist exclusion is optional, you need to obtain the blacklist BED files from ENCODE and process them to remove the prefix of “chr”. Or, you could download a blacklist from our GitHub page ([link](#)). But this does not contain the blacklists for all organisms, as for most organisms, the blacklists are still not well-curated.

Once you download the blacklists, you must keep those under a directory structure where the names of directories do not contain any white-spaces in them.

Like, they can be like this -

`/home/prince/atac_seq_project/blacklist_for_chromacs/homo_sapiens`

But not like this-

`/home/prince/atac_seq_project/blacklist_for_chromacs/homo sapiens`

Or not like this- `/home/prince/atac seq project/blacklist_for_chromacs/homo_sapiens`
(note how these directory structures have white-spaces in their names. This will affect the Genrich input of blacklists, and is an inherent limitation of how Genrich takes in the blacklist input from a directory structure.)

Step 5: Running the entire pipeline

Once you are done with the parameter setup, in step 5, you just need to click on “Run Pipeline”. This begins the entire process in a modular manner, beginning with FastQC, MultiQC, trimming by Trim Galore (if not disabled), alignment and coverage studies, peak calling and then peak annotation.

The screenshot shows the 'Step 5: Analysis' tab of a web-based pipeline interface. At the top, there are five tabs: 'Step 1: Data Setup', 'Step 2: Quality Control', 'Step 3: Reference Selection', 'Step 4: Peak Caller Selection', and 'Step 5: Analysis'. The 'Step 5: Analysis' tab is active. Below the tabs, there is a section titled 'Pipeline Instructions' with the following text: 'Click 'Run Pipeline' to generate peak files with annotation. First-time runs may take longer due to genome setup and indexing.' Below this, there is a section titled 'Output locations:' with a list: '- FastQC, MultiQC, Trimmed data: base output directory', '- Aligned BAM files and coverage: bam_output folder', '- Peak files: peak_files folder', and '- Annotated peaks: Annotated_Peaks subfolder'. Below the list, there is a statement: 'You MUST run this pipeline at least once before you perform the Differential, Motif, Footprinting Analysis'. At the bottom of the instructions section, there are two buttons: '← Back' and 'Run Pipeline'. Below the instructions section, there are two sub-sections: 'Differential Analysis' and 'Motif Analysis'. The 'Differential Analysis' sub-section has a button 'Configure DiffBind'. The 'Motif Analysis' sub-section has a button 'Configure NOISeq'.

You can even track the real-time progress of the run while the pipeline is running.

The screenshot shows a terminal window with the following output:

```
application/gzip
application/gzip
application/gzip
application/gzip
application/gzip
application/gzip
Started analysis of SRR5800658_1.fastq.gz
Started analysis of SRR2927084_1.fastq.gz
Started analysis of SRR5800657_2.fastq.gz
Started analysis of SRR5800661_2.fastq.gz
Started analysis of SRR5800657_1.fastq.gz
Started analysis of SRR2927085_1.fastq.gz
Started analysis of SRR5800662_2.fastq.gz
```

You can even observe the outputs in the required directory inside the base_output_directory, which you assigned in step 1. But in order to know where to look for which result, an architecture of the output file design can help guide you. [[link to result architecture](#)]

Step 5a: Running DiffBind

You will notice that ChromAcS step 5 has two Differential Analysis modules in step 5, which are assigned for differential binding analysis following peak annotation. When the peak calling is over, you can select either DiffBind or NOISeq for differential peak analysis.

When do you select DiffBind?

If your research design includes more than one biological replicate per condition, then we would recommend you to use the DiffBind module of ChromAcS. In our case study, there were three biological replicates per condition, that is, three replicates (SRR5800657, SRR5800658, SRR5800659) are untreated replicates, whereas, three replicates (SRR5800660, SRR5800661, SRR5800662) are treated (drug-tolerant persistors/ DTP) replicates.

How to configure DiffBind parameters?

When you click on “Configure DiffBind” a new tab will open, where you can configure the DiffBind parameters and set up the analysis-

Step 1 Step 2 Step 3 Step 4 Step 5

DiffBind Configuration

1. Select Peak Files:

Browse Peaks

FDR Threshold (0-1): 0.05

Number of Threads (default = 4): 4

Note: DiffBind is very memory intensive tool, so adjust your thread here accordingly
If run fails with error in the core setup, adjust (decrease thread count) and rerun DiffBind

Run DiffBind

Annotate Results

Note: You can skip this Control section, if there is none
Please keep in mind that these are not biological controls, rather treated.
Do NOT put untreated condition as control here

IMPORTANT: "untreated" and "treated" are symbolic labels.
Think of "untreated" as your reference or baseline condition, and "treated" as your group.
These labels can represent any two biological states — e.g.:
Untreated = Lung tissue, Treated = Kidney tissue
Untreated = Wild-type, Treated = Mutant
Untreated = Day 0, Treated = Day 5

(This DiffBind Configuration tab opens over the existing main tab, and on first look, it seems mostly empty. You need to click on the “Browse Peaks” to reveal the peak files you obtained after running the pipeline)

DiffBind Configuration

1. Select Peak Files:

SRR5800657.genrich.peak
SRR5800658.genrich.peak
SRR5800659.genrich.peak
SRR5800660.genrich.peak
SRR5800661.genrich.peak
SRR5800662.genrich.peak

Browse Peaks

Peak File Condition Replicate Control

FDR Threshold (0-1): 0.05
Number of Threads (default = 4): 4

Note: DiffBind is very memory intensive tool, so adjust your thread here accordingly
If run fails with error in the core setup, adjust (decrease thread count) and rerun DiffBind

Run DiffBind

Annotate Results

Note: You can skip this Control section, if there is none
Please keep in mind that these are not biological controls, rather technical replicates.
Do NOT put untreated condition as control here

IMPORTANT: “untreated” and “treated” are symbolic labels.
Think of “untreated” as your reference or baseline condition, and “treated” as the group.
These labels can represent any two biological states — e.g.:
Untreated = Lung tissue, Treated = Kidney tissue
Untreated = Wild-type, Treated = Mutant
Untreated = Day 0, Treated = Day 5

(Once you click on “Browse Peaks”, the peak files will appear. Still, you won’t be able to assign the conditions and replicate values to those. For that, you need to select all the peak files you would like to include in the analysis. ChromAcS does this selection-based peak analysis in differential studies because it includes two different peak callers, and you might not want to mix the peaks called by different tools in one single comparison.

DiffBind Configuration

1. Select Peak Files:

SRR5800657.genrich.peak
SRR5800658.genrich.peak
SRR5800659.genrich.peak
SRR5800660.genrich.peak
SRR5800661.genrich.peak
SRR5800662.genrich.peak

Browse Peaks

Peak File Condition Replicate Control

Peak File	Condition	Replicate	Control
SRR5800657.genrich.peak	treated	1	
SRR5800658.genrich.peak	treated	1	
SRR5800659.genrich.peak	treated	1	
SRR5800660.genrich.peak	treated	1	
SRR5800661.genrich.peak	treated	1	
SRR5800662.genrich.peak	treated	1	

FDR Threshold (0-1): 0.05
Number of Threads (default = 4): 4

Note: DiffBind is very memory intensive tool, so adjust your thread here accordingly
If run fails with error in the core setup, adjust (decrease thread count) and rerun DiffBind

Run DiffBind

Annotate Results

Note: You can skip this Control section, if there is none
Please keep in mind that these are not biological controls, rather technical replicates.
Do NOT put untreated condition as control here

IMPORTANT: “untreated” and “treated” are symbolic labels.
Think of “untreated” as your reference or baseline condition, and “treated” as the group.
These labels can represent any two biological states — e.g.:
Untreated = Lung tissue, Treated = Kidney tissue
Untreated = Wild-type, Treated = Mutant
Untreated = Day 0, Treated = Day 5

(Once you click on the peak files, they will appear in this format from which you can assign the different conditions, replicate number, and the control files; in the bottom part you can customize the FDR threshold, and assign the number of threads for DiffBind analysis.)

How do we set up the conditions and replicates?

In our case study, there were three biological replicates per condition; that is, three replicates (SRR5800657, SRR5800658, SRR5800659) are untreated replicates, whereas three replicates (SRR5800660, SRR5800661, SRR5800662) are treated (drug-tolerant persistors/ DTP) replicates. So we set up the conditions and replicates as follows-

The screenshot shows the DiffBind Configuration window. It has a title bar 'DiffBind Configuration' and standard window controls. The main area is divided into sections:

- 1. Select Peak Files:** A list of peak files is shown: SRR5800657.genrich.peak, SRR5800658.genrich.peak, SRR5800659.genrich.peak, SRR5800660.genrich.peak, SRR5800661.genrich.peak, and SRR5800662.genrich.peak. A 'Browse Peaks' button is to the right.
- Table:** A table with four columns: Peak File, Condition, Replicate, and Control.

Peak File	Condition	Replicate	Control
SRR5800657.genrich.peak	untreated	1	SRR2927084
SRR5800658.genrich.peak	untreated	2	SRR2927084
SRR5800659.genrich.peak	untreated	3	SRR2927084
SRR5800660.genrich.peak	treated	1	SRR2927085
SRR5800661.genrich.peak	treated	2	SRR2927085
SRR5800662.genrich.peak	treated	3	SRR2927085
- FDR Threshold (0-1):** A slider set to 0.05.
- Number of Threads (default = 4):** A text box with the value 4.
- Buttons:** 'Run DiffBind' (green) and 'Annotate Results' (cyan).
- Notes:**
 - 'Note: You can skip this Control section, if there is none. Please keep in mind that these are not biological controls, rather technical controls. Do NOT put untreated condition as control here.'
 - 'IMPORTANT: "untreated" and "treated" are symbolic labels. Think of "untreated" as your reference or baseline condition, and "treated" as your test condition. These labels can represent any two biological states — e.g.: Untreated = Lung tissue, Treated = Kidney tissue. Untreated = Wild-type, Treated = Mutant. Untreated = Day 0, Treated = Day 5.'
 - 'Note: DiffBind is very memory intensive tool, so adjust your thread here accordingly. If run fails with error in the core setup, adjust (decrease thread count) and rerun DiffBind.'

(Once you click on the condition box, a drop-down menu appears, from which you can assign either an untreated (baseline) or treated (test) condition. The replicate can be set up in a similar manner; the order does not matter, actually. The controls used here are technical controls, not the biological untreated condition as the control.)

What does the condition represent here?

In this condition parameter, you get two options: untreated and treated. Please keep in mind that these are **just symbolic labels**, which means-

Untreated ⇒ **baseline** condition ⇒ could be untreated samples, wild-type samples, day 0 of treatment, or even a different tissue you want to contrast against

Treated ⇒ **test condition** ⇒ could be treated samples, mutant samples, day 5 of treatment, or even another tissue you want to compare the former with

So, DiffBind will generate the differential sites and loss, gain sites accordingly. Which means that the logFoldChanges would be compared in terms of the baseline for those test conditions.

For instance, in our study, the ChromAcS DiffBind module will show the results of gain sites as those regions where peaks are enriched in the treated compared to the untreated, and the loss sites as the reverse.

Does the order of the peak files or the condition matter?

No, the order of the peak files or conditions does not matter at all. Just the name of the condition against the peak file matters. Be certain to put the correct condition against the desired peak files. Your peak files could be ordered like this and yet yield the correct result-

Peak File	Condition	Replicate	Control
SRR5800661	Treated (test)	1	SRR2927085
SRR5800658	Untreated (baseline)	1	SRR2927084
SRR5800662	Treated (test)	2	SRR2927085
SRR5800660	Treated (test)	3	SRR2927085
SRR5800657	Untreated (baseline)	2	SRR2927084
SRR5800659	Untreated (baseline)	3	SRR2927084

The results of ChromAcS DiffBind module will always consider the peak files under the untreated condition as baseline, regardless of any order your peak files are oriented in. Please do not assign the same replicate value to two different peaks.

Does it work for only three replicates?

No, the ChromAcS DiffBind module is not limited to triplicates studies. You can use this for anything greater than two per condition. This means that at least two samples per condition are essential to run this module.

Does it work if I have a single sample per condition?

No, in that case, you have to select the ChromAcS NOISEq module.

Why do we need to set up the thread again here? We already set up in step 1?

DiffBind can be very resource-intensive in terms of memory usage, and thus, the run may crash even with 8 threads if the RAM is limited. So, we would suggest using 4 threads in a 16 GB RAM system, even if the 8 threads worked fine for the main pipeline run.

How do you annotate the results of ChromAcS DiffBind analysis?

Simply by clicking the Annotate Results inside the DiffBind Configuration tab, once the DiffBind run is complete. Do NOT click Annotate Results until the DiffBind run is over.

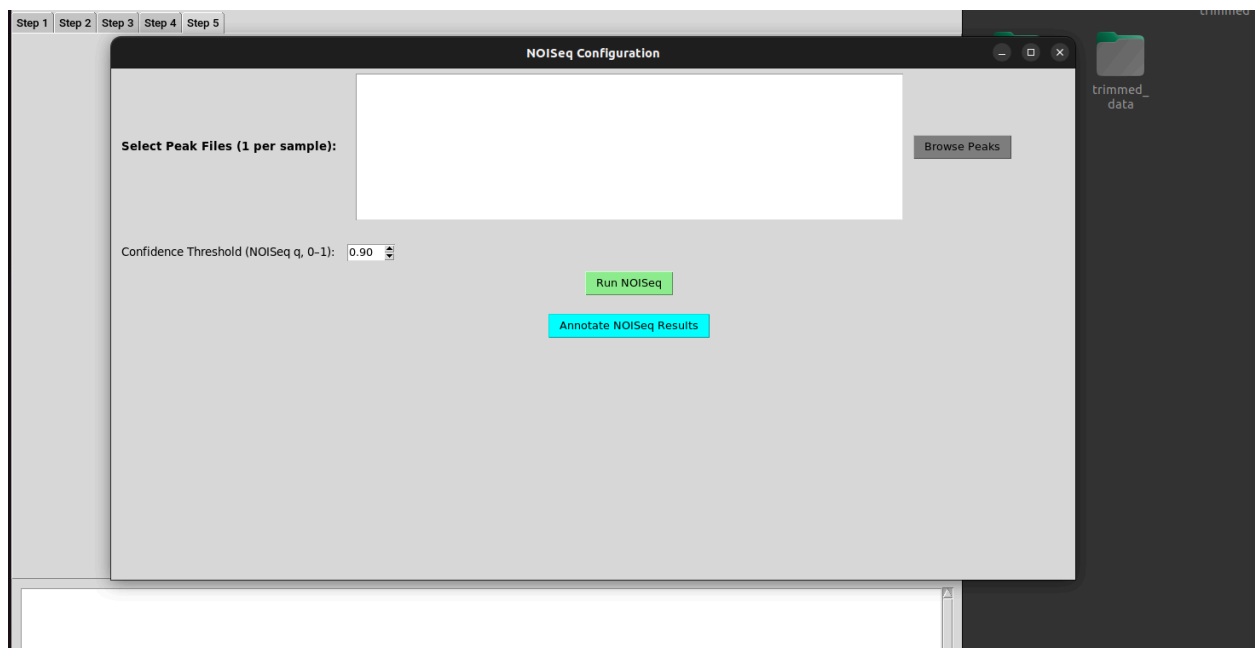
Step 5b: Running NOISeq

When do you select NOISeq?

If your research design includes only one biological sample per condition, then we would recommend that you use the NOISeq module of ChromAcS.

How to configure DiffBind parameters?

When you click on “Configure NOISeq” a new tab will open, where you can configure the NOISeq parameters and set up the analysis-



(This NOISeq Configuration tab opens over the existing main tab, and on first look, it seems mostly empty. You need to click on the “Browse Peaks” to reveal the peak files you obtained after running the pipeline)

NOISeq Configuration

Select Peak Files (1 per sample):

SRR5800657.genrich.peak
SRR5800658.genrich.peak
SRR5800659.genrich.peak
SRR5800660.genrich.peak
SRR5800661.genrich.peak
SRR5800662.genrich.peak

Browse Peaks

Peak File Condition

Confidence Threshold (NOISeq q, 0-1): 0.90

Run NOISeq

Annotate NOISeq Results

(Once you click on "Browse Peaks", the peak files will appear. Still, you won't be able to assign the conditions and replicate values to those. For that, you need to select all the peak files you would like to include in the analysis. ChromAcS does this selection-based peak analysis in differential studies because it includes two different peak callers, and you might not want to mix the peaks called by different tools in one single comparison)

NOISeq Configuration

Select Peak Files (1 per sample):

SRR5800657.genrich.peak
SRR5800658.genrich.peak
SRR5800659.genrich.peak
SRR5800660.genrich.peak
SRR5800661.genrich.peak
SRR5800662.genrich.peak

Browse Peaks

Peak File	Condition
SRR5800657.genrich.peak	treated
SRR5800658.genrich.peak	treated
SRR5800659.genrich.peak	treated
SRR5800660.genrich.peak	treated
SRR5800661.genrich.peak	treated
SRR5800662.genrich.peak	treated

Confidence Threshold (NOISeq q, 0-1): 0.90

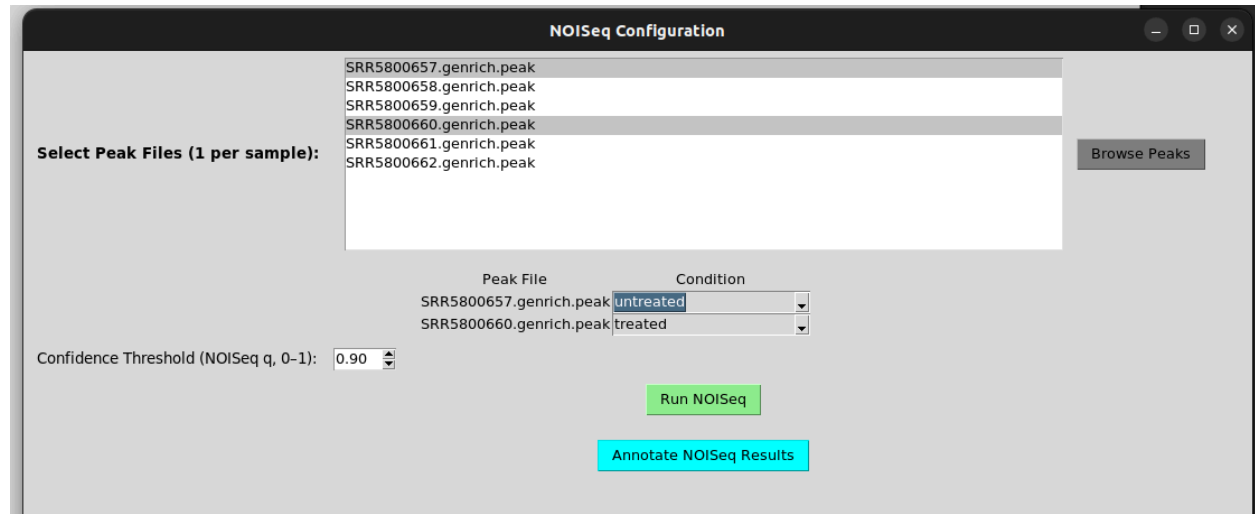
Run NOISeq

Annotate NOISeq Results

(Once you click on the peak files, they will appear in this format from which you can assign the different conditions; NOISeq does not allow control files; in the bottom part, you can customize the NOISeq confidence threshold, which is actually quite different from typical FDR values. For instance, 0.9 would mean 90% confidence, thus a lower number of differential peaks at higher values.)

How do we set up the conditions?

In our case study, there were three biological replicates per condition. But assuming that we had only one sample per condition, that is, SRR5800657 as the untreated condition and SRR5800660 as the treated condition, the configuration could be done as follows-



(Once you click on the condition box, a drop-down menu appears, from which you can assign either an untreated or treated condition)

What does the condition represent here?

In this condition parameter, you get two options: untreated and treated. Please keep in mind that these are **just symbolic labels**, which means-

Untreated ⇒ **baseline** condition ⇒ could be untreated samples, wild-type samples, day 0 of treatment, or even a different tissue you want to contrast against

Treated ⇒ **test condition** ⇒ could be treated samples, mutant samples, day 5 of treatment, or even another tissue you want to compare the former with

So, NOISeq will generate the differential sites and loss, gain sites accordingly. Which means that the logFoldChanges would be compared in terms of the baseline for those test conditions. For instance, in our study, the ChromAcS NOISeq module will show the results of gain sites as those regions where peaks are enriched in the treated compared to the untreated, and the loss sites as the reverse.

Does the order of the peak files or the condition matter?

No, the order of the peak files or conditions does not matter at all. Just the name of the condition against the peak file matters. Be certain to put the correct condition against the desired peak files. Your peak files could be ordered like this and yet yield the correct result-

Peak File	Condition
SRR5800660	Treated (test)
SRR5800657	Untreated (baseline)

The results of the ChromAcS NOISEq module will always consider the peak files under the untreated condition as baseline, regardless of any order your peak files are oriented in.

Does our NOISEq module work for studies with replicates?

No, the ChromAcS NOISEq module is not optimized to work for replicates. If you have biological replicates in your study, please use the ChromAcS DiffBind module exclusively.

Does it work if I have a single sample per condition?

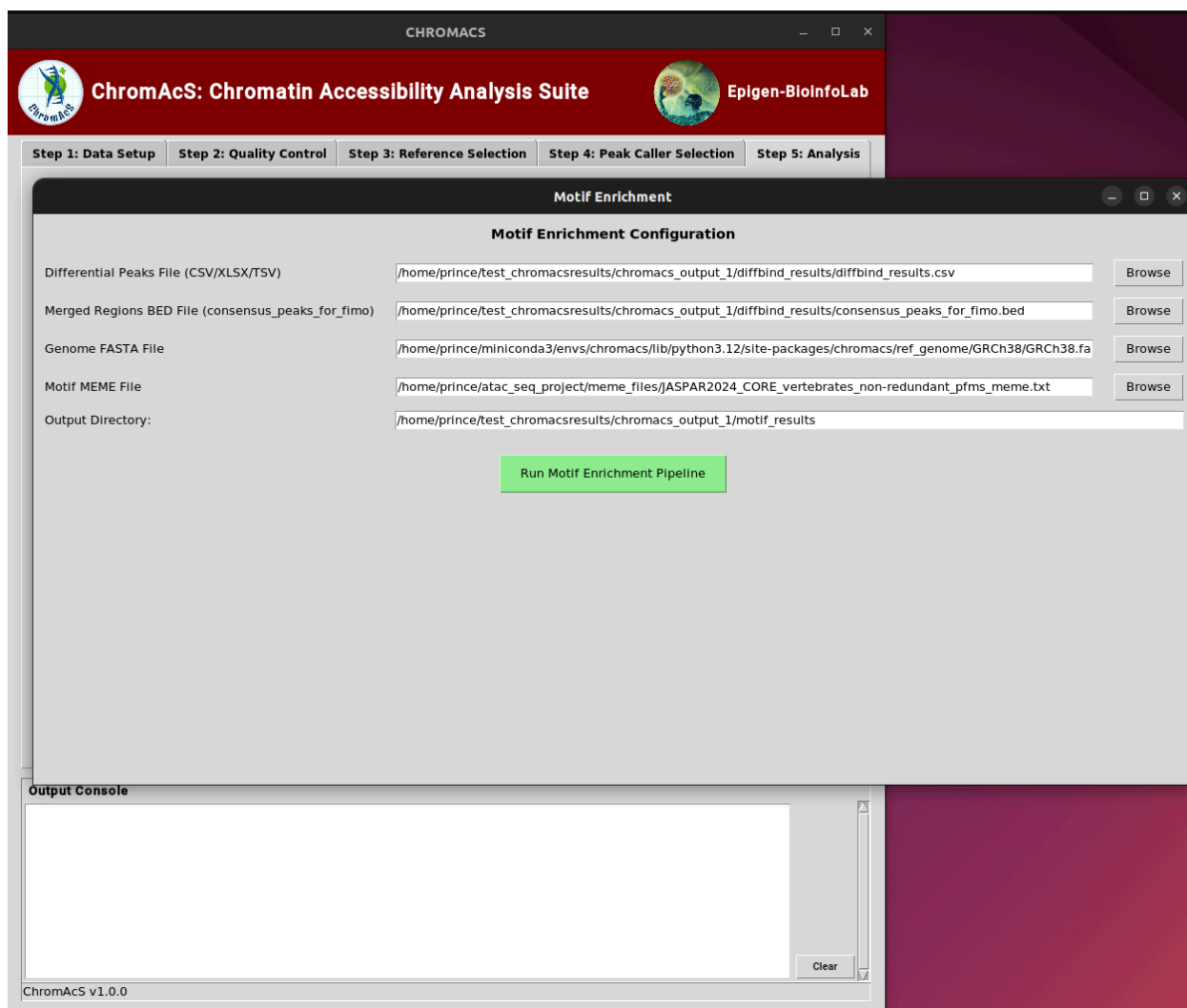
Yes, NOISEq is optimized for a single sample per condition and works on a noise-based assumption for differential peak analysis.

How do you annotate the results of ChromAcS NOISEq analysis?

Simply by clicking the Annotate Results inside the NOISEq Configuration tab, once the NOISEq run is complete. Do NOT click Annotate Results until the NOISEq run is over.

Step 5c: Motif Enrichment and Footprinting

Step 5 also includes two modules for Transcription factor motif analysis, one is motif enrichment involving mostly FIMO, and another is for motif footprinting involving TOBIAS.



The screenshot shows the ChromAcS web interface. At the top, there is a header with the logo, the text "ChromAcS: Chromatin Accessibility Analysis Suite", and "Epigen-BioInfoLab". Below the header is a navigation bar with five steps: "Step 1: Data Setup", "Step 2: Quality Control", "Step 3: Reference Selection", "Step 4: Peak Caller Selection", and "Step 5: Analysis". The "Step 5: Analysis" tab is selected. The main content area is titled "Motif Enrichment" and contains a "Motif Enrichment Configuration" section. This section has five input fields, each with a "Browse" button: "Differential Peaks File (CSV/XLSX/TSV)" with the path "/home/prince/test_chromacsresults/chromacs_output_1/diffbind_results/diffbind_results.csv", "Merged Regions BED File (consensus_peaks_for_fimo)" with the path "/home/prince/test_chromacsresults/chromacs_output_1/diffbind_results/consensus_peaks_for_fimo.bed", "Genome FASTA File" with the path "/home/prince/miniconda3/envs/chromacs/lib/python3.12/site-packages/chromacs/ref_genome/GRCh38/GRCh38.fa", "Motif MEME File" with the path "/home/prince/atac_seq_project/meme_files/JASPAR2024_CORE Vertebrates non-redundant pfms meme.txt", and "Output Directory:" with the path "/home/prince/test_chromacsresults/chromacs_output_1/motif_results". Below these fields is a green button labeled "Run Motif Enrichment Pipeline". At the bottom of the interface is an "Output Console" area, which is currently empty, with a "Clear" button on the right. The version "ChromAcS v1.0.0" is displayed in the bottom left corner.

(Configuring across motif enrichment module is pretty straightforward, first you need to input the differential peak file obtained from step 5b, or step 5c [for DiffBind it is the “diffbind_results.csv” for NOISEq it is the “noisq_results.xlsx”]), the second input is the consensus_peaks_for_fimo.bed which will also be present in the same directory if you had run either step 5b (DiffBind) or step 5c (NOISEq). The third field is usually auto-filled with the reference genome fasta sequence. For the fourth field, you must obtain a motif MEME file according to your choice and preferences. Here we have used https://jaspar.elixir.no/download/data/2024/CORE/JASPAR2024_CORE_non-redundant_pfms_jaspar.txt from JASPAR. The output directory will be auto-created. Just click Run Motif Enrichment Pipeline to begin the analysis. This may take some time.

The screenshot displays the ChromAcS web interface with the 'TOBIAS Motif Footprinting Configuration' window open. The main interface has a red header with the ChromAcS logo and 'EpiGen-BioInfoLab'. Below the header are five steps: Step 1: Data Setup, Step 2: Quality Control, Step 3: Reference Selection, Step 4: Peak Caller Selection, and Step 5: Analysis. The 'Motif Analysis' tab is selected, showing options for 'Motif Enrichment' and 'Footprinting Analysis'. The 'Footprinting Analysis' section is active, showing a 'Configure Footprinting' button. The 'TOBIAS Motif Footprinting Configuration' window is a modal dialog with the following sections:

- Baseline BAM Files (Please use name.sorted.bam):** Three input fields with 'Add Files' buttons.
- Test BAM Files (Please use name.sorted.bam):** Three input fields with 'Add Files' buttons.
- Baseline Peak Files:** Three input fields with 'Add Files' buttons.
- Test Peak Files:** Three input fields with 'Add Files' buttons.
- Motif File (.meme):** One input field with a 'Browse' button.
- Output Directory:** One input field with a 'Browse' button.
- Run Footprinting:** A green button at the bottom.

The 'Output Console' at the bottom of the main interface shows the following log:

```

Starting motif annotation...
True annotation match rate: 17.6%
Saved annotated up motifs to /home/prince/test_chromacsresults/chromacs_output_1/motif_results/motif_up_annotated.json
True annotation match rate: 5.5%
Saved annotated down motifs to /home/prince/test_chromacsresults/chromacs_output_1/motif_results/motif_down_annotated.json
Motif annotation completed successfully!
Loading json files...
Plotting for upregulated...
Plotting for downregulated...
Plotting for combined...
Plotting of motif.gene complete

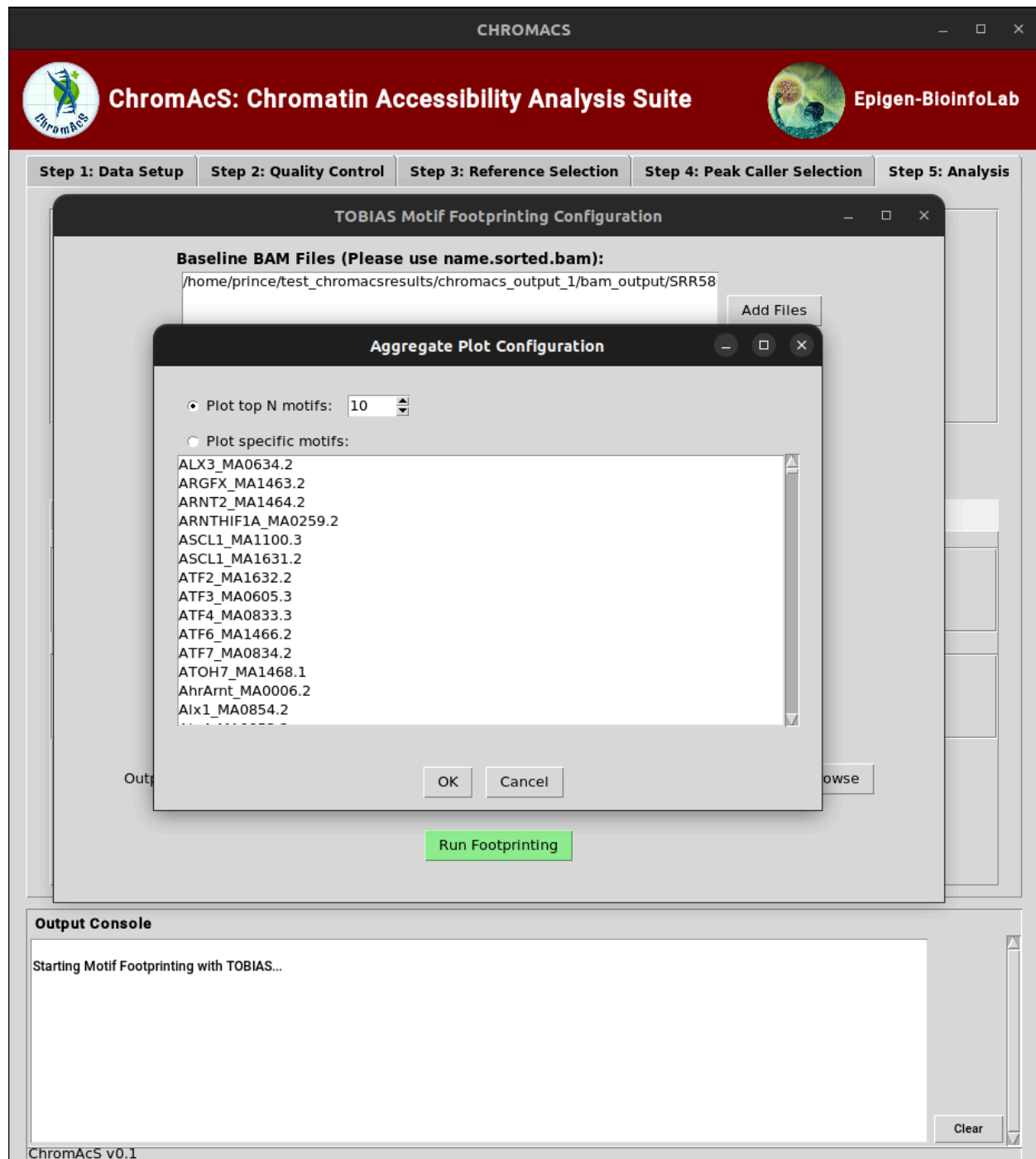
```

ChromAcS v1.0.0

(ChromAcS configures Motif footprinting of TOBIAS and integrates it into a single pipeline and provides publication-ready outputs. This can be done very easily; if you have run the pipeline up to peak calling once, then you can perform motif footprinting here. When you browse baseline BAM files, it will automatically take you to the BAM_files directory, where you can find your desired BAM files. **Be careful to choose the name_sorted_bam files.** The same goes for the test BAM files. The baseline peaks are also selected in a similar manner, just be cautious to put the consistent peak files, avoiding mixing up Genrich or MACS3 peaks. For motif meme files, we used

https://jaspar.elixir.no/download/data/2024/CORE/JASPAR2024_CORE_non-redundant_pfms_jaspar.txt; but you may choose any desired meme file.)

During the motif footprinting analysis, a dialogue box will appear mid-run and ask for the desired motifs or the top significant motifs you would like to see-



Does our motif analysis work for studies without replicates?

Yes, if you provide single samples per condition, they will be analyzed. For multiple replicates per condition, ChromAcS generates merged bam and merged peaks per condition, and process them.

Random Navigation “Hacks” in ChromAcS

ChromAcS is designed to be modular in nature, and thus, if by any chance it crashes on any local device, the user can rerun it on the same base output directory and proceed from where it crashed. But in order to track everything and outputs generated, the user needs to have a detailed overview of the output directory architecture, that is, how the outputs are saved and in which directories and sub-directories.

A typical structure of the output files would look like this-

base_output_directory/

- |— **fastqc_raw/** → **Raw FastQC reports**
- |— **multiqc_raw/** → **MultiQC summary (raw)**
- |— **trimmed_data/** → **Trimmed FASTQs + reports**
- |— **fastqc_trimmed/** → **FastQC reports of trimmed data**
- |— **multiqc_trimmed/** → **MultiQC summary of trimmed data**
- |— **bam_output/** → **Sorted BAMs (name-sorted and coordinate-sorted) + index (of sorted BAM files) + flagstat results**
- |— **normalized_coverage/** → **BigWig files of the sorted BAM**
 - |— **Coverage_profiles/** → **TSS enrichment heatmap based on the BigWIG files**
- |— **peak_files/** → **Peak files from Genrich/MACS3**
 - |— **Annotated_Peaks/** → **Annotated peaks + peak statistics summary**
- |— **diffbind_results/** → **DiffBind differential analysis results**
 - |— **Annotated_DiffBind/** → **Annotated files + statistics summary**
- |— **noisq_results/** → **NOISeq differential analysis results**
 - |— **Annotated_NoISeq/** → **Annotated files + statistics summary**
- |— **motif_results/** → **Motif enrichment analysis results + plots**
 - |— **motif_gene_analysis/** → **motif-to-gene association + plots**
- |— **motif_footprinting/** → **Motif footprinting results**
 - |— **bindetect_results/** → **all on differential motif footprinting across two conditions**
 - |— **tobias_aggregate_plots/** → **shows the differential footprints as plots**

Here, all these directories are auto-generated once ChromAcS begins to run (i.e., when the user clicks run pipeline, however, diffbind_results and noisq_results would be generated later when the user runs those modules. Some of the directories contain sub-directories within which the relevant results of the files processed in the directory are saved (the sub-directories are shown here under the main directories).

Can I check the results once one step is done but ChromAcS is still running?

Yes, you can absolutely check the results in the directories while ChromAcS is still running. For instance, if the MultiQC step is shown to be complete within the real-time text widget update on the ChromAcS screen, you can navigate to the relevant directory and check the outputs while trimming is still running inside ChromAcS.

How is ChromAcS modular, and how can I exploit this?

ChromAcS proceeds the pipeline beginning from FastQC till peak calling and annotation seamlessly, after which the user can proceed for differential peak analysis using DiffBind or NOISeq ChromAcS module. So, it may seem that only the differential peak analysis is modular, and the rest of the steps are not. But in reality, ChromAcS can recognize which steps have been performed earlier within the main pipeline and continue to run the pipeline from the subsequent steps. Let's say that for some unexpected reason, ChromAcS crashed in the local device during the trimming of samples. Now, if the user reruns ChromAcS and, while navigating, assigns the same base output directory, it will be able to continue from trimming. Even if the trimming fails during, let's say the third out of eight samples we ran, ChromAcS will proceed to run trimming during the rerun from the third sample.

How is the modular nature of ChromAcS helpful during parameter optimization?

The modular nature of ChromAcS can be significantly effective during parameter optimization of the analysis. Let's assume that the user needs to observe which d-value of Genrich can be most relevant to achieve the best biological rationale of their results, or which p-value they want for their analysis.

In that case, the user can run Genrich with one d-value (for example, 100) during the first run. The output peak files will be saved in the peak files directory. Now the user can check those peak files and perform differential analysis through the DiffBind or NOISeq modules.

Now, for a different d-value set up (for example, 200) during their second run, the user first needs to cut the peak files directory only from their base output directory and paste it to a different directory (if they wish to keep this process non-destructive). And then rerun ChromAcS with their desired d-value of 200, and during this second run, ChromAcS will begin from the peak calling directly due to its being modular in nature. This significantly helps to reduce time during parameter optimization.

Users can employ a similar approach to rerun ChromAcS for different peak callers (for example, trying with Genrich for the first run, and then MACS3 for another run), for differential analysis (using DiffBind with different thresholds of confidence), and several other combinations of parameters.



ChromAcS-AddOn: Additional Analysis Tool - User Manual

Tab 1: Peak Overlap Analysis

This module checks whether ATAC-seq peaks (from ChromAcS or DiffBind) overlap with genomic regions such as histone marks or enhancer annotations.

Inputs:

- One ATAC-seq peak file (ATAC-seq peak, Differential ATAC-seq file, or annotated file). This file should have the first three columns: chromosome, start, and end. Only tab-separated text file with header.
- Upload up to 5 additional BED/peak files. Only tab-separated text file with no header (or, add a hast tag # at the beginning if your tabular file has a header). This file should have the minimum first four columns in the order: chromosome, start, end, name. Rest columns can be in any order. Example:

Example-1 (BED, no header):

```
chr1 1500 1800 H3K27ac_peak1
chr1 3500 3800 H3K27ac_peak2
```

Example-1 (Peak, # tag header):

```
#chromosome start end name
chr1 100036623 100037420 H3K4me1
chr1 100047999 100048530 H3K4me1
chr1 100049270 100049895 H3K4me1
```

- Set Window size in bp (e.g., 200). This will extend the ATAC-seq peaks on both sides for this base-pair to find overlap.

Steps:

1. Click 'Browse' to upload the ATAC-seq peak file.
2. Enter the window size (e.g., 200).
3. Upload up to 5 BED files containing enhancer/histone data.
4. Select an output file name.

5. Click 'Run Overlap Analysis'

What it does:

- Cleans and formats the ATAC-seq file
- Extends the peaks by the specified window
- Normalizes chromosome naming (chr prefix)
- Uses bedtools intersect to find overlapping regions
- Outputs a combined overlap file

Output Files

- Overlap.tsv: Contains ATAC-seq peaks with overlaps against BED files

Screenshot: Peak Overlap Analysis Tab

The screenshot shows the 'ChromAcS-AddOn: Additional Analysis Toolkit' window. The title bar includes the text 'ChromAcS-AddOn: Additional Analysis Toolkit' and standard window controls. The interface has a red header bar with the ChromAcS logo on the left and the 'Epigen-BioinfoLab' logo on the right. Below the header, the 'Peak Overlap Analysis' tab is selected, and the 'Compare with Expression' sub-tab is active. The main area contains several input fields and buttons: 'ATAC-seq Peaks File:' with a 'Browse' button, 'Window Size (bp):' with a text box containing '200', and five 'BED File' inputs (1-5), each with a 'Browse' button. At the bottom, there is an 'Output File:' input with a 'Browse' button and a large blue 'Run Overlap Analysis' button. The status bar at the very bottom indicates 'Ready'.

ChromAcS-AddOn: Additional Analysis Toolkit

ChromAcS-AddOn: Additional Analysis Toolkit

Epigen-BioinfoLab

Peak Overlap Analysis | Compare with Expression

ATAC-seq Peaks File: Browse

Window Size (bp):

BED File 1: Browse

BED File 2: Browse

BED File 3: Browse

BED File 4: Browse

BED File 5: Browse

Output File: Browse

Run Overlap Analysis

Ready

Tab 2: ATAC-seq Compare with Expression

This module merges annotated ATAC-seq peaks with gene expression data.

Inputs:

- Annotated ATAC-seq file (from ChromAcS). Only tab-separated text file WITH header.
- Gene expression matrix (tab-separated text file with header line)
- Gene ID column number in each file.

Example (RNA_expression.tsv):

ID	logFC	AveExpr	t	P.Value	adj.P.Val	B
PEG3	13.14	0.94	96.70	0.001	0.002	19.02
RP11-572P						
18.1	11.83	0.29	66.56	0.001	0.002	17.19
DCT	8.12	2.67	66.14	0.001	0.002	18.80
NEFM	6.47	5.43	63.65	0.001	0.002	19.41

Steps:

1. Upload the annotated ATAC-seq file (or annotated differential ATAC-seq file)
2. Upload the expression file.
3. Enter the matching gene ID (or gene name/symbol) column number for both files.
4. Select an output file.
5. Click 'Run Expression Merge'.

What it does:

- Matches gene IDs across both files
- Appends expression values to the peak annotation
- Handles missing or unmatched values with 'NA'

Output Files

- Combined_ATACseq_Expression.tsv: Merged table of peaks + gene expression

Screenshot: Compare with Expression Tab

Peak Overlap Analysis: Compare with Expression

Annotated ATAC-seq File:

Expression Data File:

ATAC-seq ID Column:

Expression ID Column:

Output File:

Ready

Troubleshooting

- Ensure all files are tab-delimited.
- Use a similar format for chromosome name.
- Ensure bedtools is installed and in the system PATH.
- For large files, allow a few minutes to complete.

Support

Contact: epigen.bioinfo@lab@gmail.com
info@epigen-bioinfo.com

Website: www.epigen-bioinfo.com