Improved Visualization and Scalability of ChIP-Seq Quality Control Analysis

National Institute of Diabetes and Digestive and Kidney Diseases

Jacob Vieira, **Ryan Dale, Elissa Lei** | Laboratory of Cellular and Developmental Biology

Problem

Chromatin Immunoprecipitation Sequencing (ChIP-seq) is a type of next generation sequencing that profiles a protein's binding sites across the genome. The R package ChIPQC is a powerful tool for performing quality control (QC) analysis of ChIP-seq data. However, the package's standard functions lack the ability to visualize this data well. This default reporting is passable when the scope of the analysis is small, but becomes effectively unreadable in analyses with large numbers of samples. This default reporting also comes with limited options for plot customization. Adding or removing even a single library means the entire analysis process needs to be re-run to create a new visualization. These issues make use of the package suboptimal for large-scale analyses of ChIP-seq data.

Goal

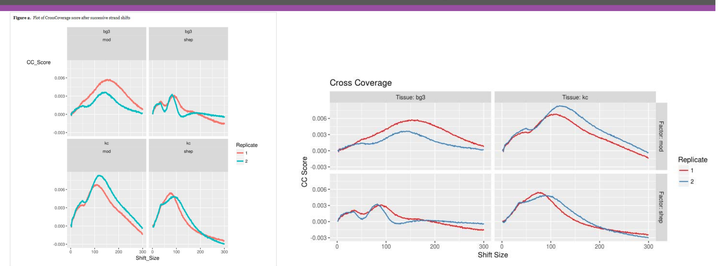
The target features for an improved ChIPQC script include:

1) Support for large experiments.
2) Highly customizable report generation.
3) Eliminate need for redundant re-processing of samples.
4) Parallelization capability for cluster computing.
5) Parameter interface for re-use without changing code.

Solution

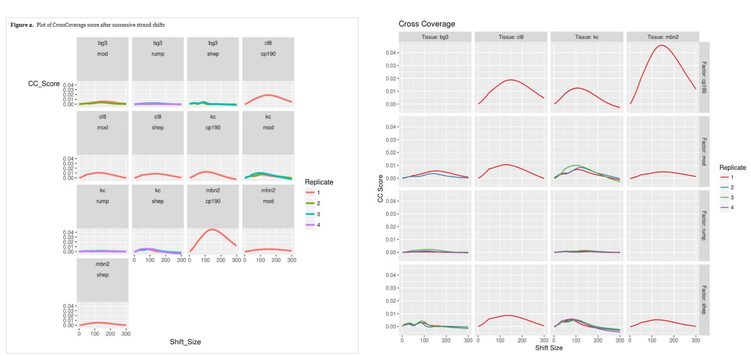
Achieving the desired functionality with ChIPQC was not possible using strictly the package's out of the box code and functions. Fortunately, ChIPQC is an open source package, meaning its internal code is publicly accessible via GitHub. Using this source code, the reporting functions were reverse engineered in a parameterizable R Markdown. Even changing just a few lines of code can allow for radical alterations in functionality. This allowed for the implementation of a whole host of new and improved features, including all of the target features.

Standard vs. Improved



A. Standard Small Plot. *B.* Improved Small Plot.

Figure 1. For a small scale analysis containing only 4 distinct sample categories, the standard script produces a passable plot. (Refer to **Figure 6** for information on cross coverage plots.)



A. Standard Large Plot. *B.* Improved Large Plot.

Figure 2. For a larger analysis containing 13 distinct sample categories, the standard visual makes poor use of space. The combination of vertically compressed plots and thick lines makes it difficult to read, and lack of two-dimensional faceting makes comparisons difficult. These issues are remedied by the improved plot.

Workflow

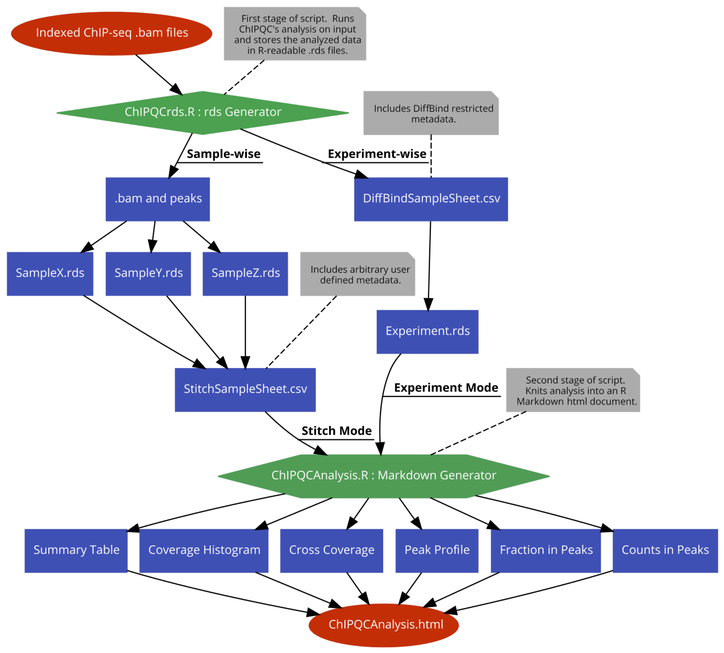


Figure 3. Workflow diagram for improved ChIPQC scripts. These scripts can operate on samples in two distinct ways. The first is the custom coded sample-wise path, running analysis on individual samples and stitching the analyzed data together later in the pipeline. The second is the reverse-engineered experiment-wise path, running analysis on multiple samples (an "experiment") together. Both methods can be parallelized, but the former has less redundancy and greater flexibility.

Command Line Interface

```
Stage 1: ChIPQCrds.R
Usage: Rscript ChIPQCrds.R  [-h] [-i  INPUT] [-p  PEAKS] [-o  OUTPUT]
                            [-c  CHROMOSOMES…] [-w  WORKERS]

Arguments:
-h                          Displays help message and exits.
-i  INPUT                   Path to input sample reads (.bam) or DiffBind
                            sample sheet (.csv) to analyze.
-p  PEAKS                   Path to called peaks for sample-wise reads.
-o  OUTPUT                  Path to output rds file to generate.
-c  CHROMOSOMES…            Space separated list of chromosomes to analyze.
                            Analyzes all if none specified.
-w  WORKERS                 Number of worker CPUs for experiment-wise
                            analysis.

Stage 2: ChIPQCAnalysis.R
Usage: Rscript ChIPQCAnalysis.R [-h] [-i INPUT] [-o OUTPUT] [-t TITLE] [-e ECHO]
                                [-x FACETX] [-y FACETY] [-z FACETZ] [-p PALETTE]
-h                          Displays help message and exits.
-i  INPUT                   Path to stitch sample sheet (.csv) or analyzed
                            experiment (.rds) to analyze.
-o  OUTPUT                  Path to output html file to generate.
-t  TITLE                   In-document title for analysis markdown.
-e  ECHO                    Echoes script code in generated markdown.
-x  FACETX                  Factor to facet columns of plots by.
-y  FACETY                  Factor to facet rows of plots by.
-z  FACETZ                  Factor to color plots by.  X axis for reads in
                            peaks plots.
-p  PALETTE                 Qualitative RColorBrewer palette to color with.
```

Figure 4. Usage syntax and parameter descriptions for command line interface. This interface makes scripts highly customizable and reusable without any code changes, and allows quality control analysis to be easily performed on command-line only computer systems such as the NIH's Helix and Biowulf clusters.

Code Access

The improved ChIPQC scripts are publicly available for download on GitHub as **ReplexChIPQC**.

Repository:
https://github.com/jmvieira97/ReplexChIPQC

Shortlink:

https://git.io/v7PqR

Quality Control Analysis

The improved ChIPQC analysis pipeline takes a collection of ChIP-seq bam files and called peaks and produces a table of summary information and five figures in each of its reports. This example compares 8 *Drosophila* ChIP-seq libraries. "Tissues" BG3 and Kc are cell types; "Factors" Mod(mdg2.2) and Shep are proteins that have been immunoprecipitated in these samples.

ChIPQC Summary Table

SampleID	Tissue	Factor	Replicate	Peaks	Reads	Map%	Filt%	Dup%	ReadL	FragL	RelCC	SSD	RiP%
bg3_mod_1	bg3	mod	1	10402	23790575	100	0	75.60	50	179	2.64	2.72	11.300
bg3_mod_2	bg3	mod	2	2186	13533395	100	0	76.20	50	165	2.62	2.75	5.790
bg3_shep_1	bg3	shep	1	5379	4176250	100	0	48.20	36	95	1.54	1.28	11.200
bg3_shep_2	bg3	shep	2	350	6782227	100	0	40.40	36	84	1.77	1.11	0.469
kc_mod_1	kc	mod	1	2904	16009718	100	0	70.70	50	102	1.98	5.92	14.000
kc_mod_2	kc	mod	2	3981	14256129	100	0	68.00	50	132	1.99	5.41	16.200
kc_shep_1	kc	shep	1	3201	7061661	100	0	7.38	36	74	1.63	1.03	3.990
kc_shep_2	kc	shep	2	1880	12097312	100	0	9.02	36	96	1.35	1.13	2.300

Table 1. Summary table concisely displaying metrics and metadata for samples. Peaks column has been added, path columns have been removed, and display has been made cleaner.

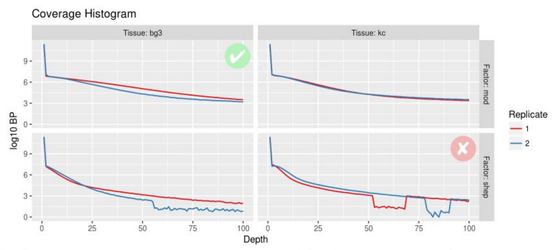


Figure 5. Coverage histogram plotting log10 of base pairs of the genome against read depth. Good samples have greater numbers at higher depths, and contain little noise.

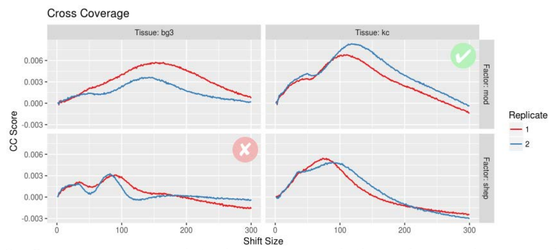


Figure 6. Cross coverage plot showing correlation of opposite strands' reads. The maximum is the size of the protein binding site in base pairs. Good samples have a "phantom" peak followed by a significantly higher true peak.

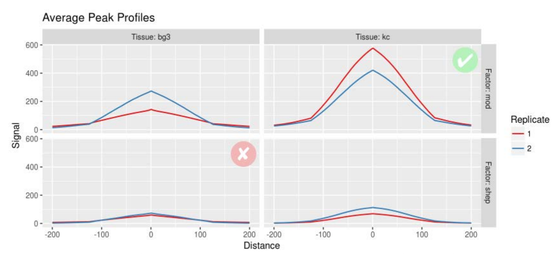


Figure 7. Peak profile plot showing the average signal profile of all called peaks for a sample. Noise will appear in samples with small numbers of called peaks. Good samples will have higher maximum signal and smoother profiles.

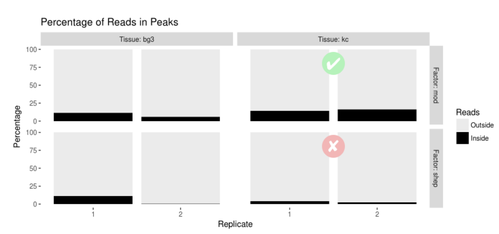


Figure 8. Stacked bar plot showing percentage of reads inside and outside of peaks. Good samples will have higher proportions of reads inside of peaks.

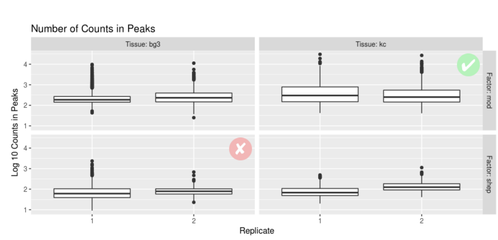


Figure 9. Box and whisker plot showing log10 of total count of reads in each peak. Log scale has been added to improve plot readability. Good samples will have counts that cluster higher.