Analyzing microbial ChIP-Seq data with Pique

R.Y. Neches [1,3]*, E.G. Wilbanks [1,3] and M.T. Facciotti [2,3]

[1]Microbiology Graduate Group, University of California, Davis.
[2]Department of Biomedical Engineering, University of California, Davis.
[3]Genome Center, University of California, Davis.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation:
Most ChIP-Seq peak finders are designed to identify protein-DNA binding events in eukaryotic datasets. To make cost-effective use of current sequencing capacity, the peak finders must be cleverly optimized to work with sparse-coverage data, and must take into account the effect of chromatin structure on the variation in background coverage. While numerous effective peak finders have been developed for eukaryotic systems, the approaches used can be suprisingly error prone in our hands when run on high-coverage bacterial and archaeal ChIP-Seq datasets.

Results:
Fortunately, many of the statistical challenges for peak detection inherent in eukaryotic ChIP-seq data are not present in bacterial and archaeal datasets. This is due in part to higher genome coverage (typically in inverse proportion to genome size) and in part to the absence of non-random coverage variation from to highly structured chromatin. We have developed Pique, a conceptually simple, easy to use ChIP-Seq peak finding application for bacterial and archaeal ChIP-Seq data. The software is cross-platform and Open Source, and based on Open Source dependencies. Output is provided in standard GFF files, and easily imported into the Gaggle Genome Browser for manual curation and data exploration.

Availability:
The software is available under the BSD-3 license at
http://github.com/ryneches/pique.
A tutorial and test data are included with the documentation.

Contact: ryneches@ucdavis.edu

1 INTRODUCTION

Next generation sequencing coupled with chromatin immunoprecipitation (ChIP-Seq) is revolutionizing our abilty to genomically map protein-DNA associations for a wide variety of proteins. The growing popularity of ChIP-Seq has spurred the development of over thirty peak picking algorithms (for a nearly completely list see [10]). The relative performance of represetative peak detection algorithms on eukatyoric data and methods to assess performance have been recently reviewed by several authors [6, 3, 9, 2, 7]. Many of these peak picking methods have employed a number of sophisticated strategies to detect peaks in the typically sparsely covered eukaryotic datasets for which they are designed.

While ChIP-Seq has been predominantly used to interrogate protein-DNA interactions in eukaryotic systems, there are clear advantages to adopting this technology for studying microbial systems. Microbial genomes typically are much smaller than eukaryotic genomes (the genome of *E. coli* is ≈2000 times smaller than the human genome), and have simpler replicon and chromatin structure. We were surprised to find that software that works well in eukaryotic systems tends to fail when presented with an presumably lesser challenge.

To our knowledge, only one other peak finding package, CSDeconv [5], has been explicitly developed for finding peaks in microbial ChIP-Seq data. This MATLAB package successfully finds peaks in microbial ChIP-Seq data, but its application is limited by its dependency on costly proprietary software, slow performance, lack of support for manual curation, and high false negative rate. Herein we describe Pique, a conceptually simple, Python-based peak finding package that enables easy and rapid peak finding in bacterial and archaeal ChIP-Seq datasets.

2 APPROACH

ChIP-Seq in bacteria and archaea yields coverage several orders of magnitude larger than in eukaryotic systems. This generates data with continuous signal across the microbial chromosome rather than the sparse coverage typically present in eukaryotic ChIP-Seq data. This feature of microbial ChIP-Seq data permits simpler, faster algorithms to be used. We have based our algorithm on classic noise reduction techniques from signal processing.

Pique is designed for use in systems that have genomic complexities such as IS elements, gene dosage polymorphisms and accessory genomes that cause variations in sequence coverage unrelated to ChIP, or in cases where the organism under study is not identical to the reference genome. The resulting enrichment "pedestals" and "holes" can be very problematic for detecting peaks and calculating enrichment levels. Pique allows the user to optionally supply a map of these features as a GFF file, and the software will automatically perform a segmented analysis.

Pique has integrated curation support through the Gaggle Genome Browser. This permits interactive curation of the peak list and analysis of the ChIP-Seq data in the context of other Gaggle-enabled resources. Interactive curation of a microbial ChIP-Seq data set can typically be completed in a few minutes.

*to whom correspondence should be addressed

3 METHODS

Prior to running Pique, Illumina (Solexa) reads should be quality filtered, quality trimmed, and aligned to a reference genome using the user's preferred short-read sequence aligner. Pique requires a BAM file as input [4]. We suggest using all contigs of the reference genome as the mapping target, but the user may prefer to proceed with one contig at a time if desired.

The user may optionally supply a map of coverage features. Pique supports three features; analysis regions, exclusion regions, and normalization regions. By default, Pique treats each contig as a single analysis regions, but the user may designate regions within a contig for separate analysis. This is useful where a gene dosage polymorphism has systematically altered the coverage level in a large region. Exclusion regions are simply masked out of their respective analysis regions, and are useful for removing coverage variation due to repetitive DNA. If the user designates normalization regions within an analysis region, Pique will use them to compensate for total coverage discrepancies between the background and ChIP alignments.

The user launches the primary analysis stage by providing alignment an file for the ChIP data, an alignment file for the control data, and optional an coverage feature map. The primary analysis proceeds thusly :

- The alignment files are digested, and the analysis regions are initialized. If a coverage feature map provided, the analysis regions are separated and the exclusion regions applied.
- The coverage noise threshold is measured by comparing the relative total coverage within the normalization regions. (The user should choose normalization regions that contain neither peaks nor coverage level aberrations.)
- High-k noise in coverage is removed using a Wiener-Kolmogorov filter. The filter delay α is chosen to approximate to the expected footprint size of the targeted protein. The choice of filter implies the existence of two inputs; a "true" signal, and a noise source. Both are assumed to be stationary stochastic processes combined additively.
- A Blackman window of a diameter equal to the read length is convolved with the filtered coverage track to remove features smaller than one read.
- The noise threshold in the ChIP coverage track is measured by comparing the coverage distribution in the ChIP track to the control track within user-annotated non-peak regions. Features that exceed the noise threshold are identified.
- To determine if a feature corresponds to a binding event, we require that the stop coordinate of the forward strand enrichment envelope must fall between the coordinates of the reverse strand enrichment envelope, and that the start coordinate of the reverse strand enrichment envelope fall between the coordinates of the forward strand enrichment envelope. (We call this the overlap criterion.)

For each putative peak, the enrichment ratio of the ChIP alignment to the the control alignment, estimates the binding coordinate, and reports to the user, as well as the enrichment normalization factor for that analysis region are calculated and compiled into user output. Determination of the statistical significance of a peak is highly specific to the experiment, and so Pique does not undertake this calculation.

4 DISCUSSION

The coverage enrichment is distributed differently between the forward and reverse strands due to the constrained read orientation imposed by the fragment size.

Pique does not attempt to filter peaks that are statistically insignificant. We have found that this part of the analysis is rather specific to the data and to the experiment. Pique is designed to achieve a low false-negative rate, which comes at the cost of a relatively high false-positive rate. Thus, some kind of post-filtering is necessary. Pique provides the user with output that can be used to support a variety of such statistical tests.

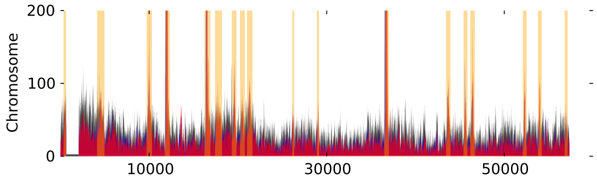

Fig. 1. Peaks found in the chromosome of the included sample dataset, derived from ChIP-seq of tfbD in *Halobacterium salinarum* sp. NRC1. Blue and red indicate coverage of reads aligned from the ChIP-derived data to the forward and reverse strands of the genome, respectively. Black indicates coverage aligned from the whole cell extract data. Orange indicates a detected peak. The maximum coverage in this data is 3204, but is shown here truncated at 200.

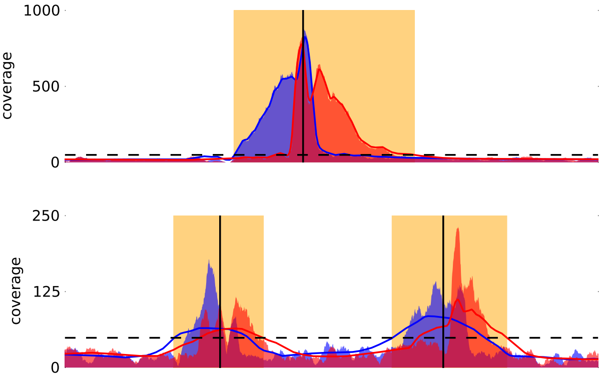

Fig. 2. Examples of large and medium-size peaks.

Some recommended filtering might include :

- Eliminate peaks that are significantly narrower than the size range of the sequencing library.
- Eliminate peaks with a normalized enrichment ratio below unity.
- Eliminate peaks that have predicted binding sites that are very skewed from the center of the enriched region.

Depending on how many peaks are recovered, the user may wish to try one or all of these, perhaps with clustering. However, if a "perfect" peak list is required, we do not recommend relying on statiastical refinements alone. To facilitate manual curation, Pique outputs a track file of the coverage, a quantitative positional data of the estimated binding sites, and a bookmark file annotating the peaks. These files are simple to process by a variety of tools, and can be loaded directly into the Gaggle Genome Browser.[1]

[1] See supplementary figure.

5 CONCLUSION

We note that Pique should also work well with eukaryotic datasets provided they are gathered with greater coverage than has been previously reported.

ACKNOWLEDGEMENT

Funding: This project was funded by UC Davis startup funds to MTF, NSF graduate fellowship AWARD NUMBER to EGW and DARPA AWARD NUMBER TO RN.

REFERENCES

[1]J. Christopher Bare, Tie Koide, David Reiss, Dan Tenenbaum, and Nitin Baliga. Integration and visualization of systems biology data in context of the genome. *BMC Bioinformatics*, 11(1):382+, July 2010.

[2]Xin Feng, Robert Grossman, and Lincoln Stein. PeakRanger: A cloud-enabled peak caller for ChIP-seq data. *BMC Bioinformatics*, 12(1):139+, 2011.

[3]Teemu Laajala, Sunil Raghav, Soile Tuomela, Riitta Lahesmaa, Tero Aittokallio, and Laura Elo. A practical comparison of methods for detecting transcription factor binding sites in ChIP-seq experiments. *BMC Genomics*, 10(1):618+, December 2009.

[4]Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics (Oxford, England)*, 25(16):2078–2079, August 2009.

[5]Desmond Lun, Ashley Sherrid, Brian Weiner, David Sherman, and James Galagan. A blind deconvolution approach to high-resolution mapping of transcription factor binding sites from ChIP-seq data. *Genome Biology*, 10(12):R142+, December 2009.

[6]Shirley Pepke, Barbara Wold, and Ali Mortazavi. Computation for ChIP-seq and RNA-seq studies. *Nature Methods*, 6(11s):S22–S32, October 2009.

[7]Morten B. Rye, Pål Sætrom, and Finn Drabløs. A manually curated ChIP-seq benchmark demonstrates room for improvement in current peak-finder programs. *Nucleic Acids Research*, 39(4):e25, March 2011.

[8]Paul Shannon, David Reiss, Richard Bonneau, and Nitin Baliga. The Gaggle: An open-source software system for integrating bioinformatics software and data sources. *BMC Bioinformatics*, 7(1):176+, 2006.

[9]Adam M. Szalkowski and Christoph D. Schmid. Rapid innovation in ChIP-seq peak-calling algorithms is outdistancing benchmarking efforts. *Briefings in Bioinformatics*, November 2010.

[10]Elizabeth G. Wilbanks and Marc T. Facciotti. Evaluation of Algorithm Performance in ChIP-Seq Peak Detection. *PLoS ONE*, 5(7):e11471+, July 2010.