Bioinformatics, 31(14), 2015, 2384–2387
doi: 10.1093/bioinformatics/btv166
Advance Access Publication Date: 19 March 2015
Applications Note

Genome analysis

iTagPlot: an accurate computation and interactive drawing tool for tag density plot

Sung-Hwan Kim[1], Onyeka Ezenwoye[2], Hwan-Gue Cho[1], Keith D. Robertson[3] and Jeong-Hyeon Choi[4,5,*]

[1]School of Computer Science and Engineering, Pusan National University, Busan, South Korea, [2]Hull College of Business, Georgia Regents University, Augusta, GA, USA, [3]Department of Molecular Pharmacology and Experimental Therapeutics, Center for Individualized Medicine, Mayo Clinic, Rochester, MN, USA, [4]Cancer Center and [5]Department of Biostatistics and Epidemiology, Medical College of Georgia, Georgia Regents University, Augusta, GA, USA

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on December 27, 2014; revised on February 19, 2015; accepted on March 16, 2015

Abstract

Motivation: Tag density plots are very important to intuitively reveal biological phenomena from capture-based sequencing data by visualizing the normalized read depth in a region.

Results: We have developed iTagPlot to compute tag density across functional features in parallel using multicores and a grid engine and to interactively explore it in a graphical user interface. It allows us to stratify features by defining groups based on biological function and measurement, summary statistics and unsupervised clustering.

Availability and implementation: http://sourceforge.net/projects/itagplot/.

Contact: jechoi@gru.edu and jeochoi@gmail.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

ChIP-seq, MBD-seq and MeDIP-seq are powerful and standard sequencing technologies to identify transcription factor binding, histone modification and DNA methylation by counting the number of reads in base resolution, i.e. tag density. One of the fundamental analyses is to visualize tag density of sequencing samples on genome browsers, e.g. the UCSC genome browser (Kent *et al.*, 2002)and IGV (Robinson *et al.*, 2011), because tag density reflects the degree of enrichment of biological aspects. Although such visualizations can clearly show the distribution of tag density in a local region, they cannot show the overall distribution of tags around genomic features. To this end, several programs and libraries in Perl, Python, R and Java have been developed. Cistrome provides correlation analyses to downstream genome feature association, gene expression analyses and motif discovery based on Galaxy workflow system (Liu *et al.*, 2011). Spark clusters tag density of regions of interest and visualizes a heatmap of clusters or regions (Nielsen *et al.*, 2012). Very recently, ngs.plot has been developed for quick mining and visualization of tag density in the command line interface (Shen *et al.*, 2014). However, some of them require programming skills or experience in the command line interface. Some have simple functions to generate plots, and do not support various graphical displays and editing, flexible grouping, fine-tuning of graphical parameters, feature identifier conversion or multiple samples, as shown in Table 1 (Anders *et al.*, 2014; Dale *et al.*, 2014; Liu *et al.*, 2011; Nielsen *et al.*, 2012; Shen *et al.*, 2014; Shin *et al.*, 2009; Statham *et al.*, 2010; Ye *et al.*, 2011).

In this article, we present iTagPlot to accurately compute and interactively visualize tag density in a graphical user interface. Like ngs.plot (Shen *et al.*, 2014), iTagPlot computes and draws the average tag density of all features for each sample. In addition, iTagPlot visualizes the tag density of individual features of interest and groups of features based on quantitative values such as gene expression, DNA methylation, CpG density and even quantiles of quantitative values. It supports parallel computation using multithreading or a grid engine and allows customizing drawing properties easily in a user-friendly interface. Given that tag density plots are the mainstay of manuscripts describing epigenomics data based on next-gen

Table 1. Program comparison where CUI and GUI stand for command line and graphical user interface, respectively

Feature	CEAS	Repitools	HTSeq	metaseq	seqMINER	Cistrome	Spark	ngs.plot	iTagPlot
Interface	CUI	CUI	CUI	CUI	GUI	Web	GUI	CUI/Web	CUI/GUI
Language	Python	R	Python	Python	Java	?	Java	Perl/R	Java/Perl
One end[a]	✓	✓	✓	✓	✓	✓		✓	✓
Both ends[a]	✓			✓		✓		✓	✓
Line chart	✓	✓	✓	✓	✓			✓	✓
Area chart									✓
Heatmap		✓		✓	✓	✓	✓	✓	✓
Functional group									✓
Quantity-based group		✓				✓			✓
Clustering group					✓	✓	✓		✓
Identifier conversion									✓
Multiple samples					✓		✓	✓	✓
Graphical editing									✓

[a]Function to visualize one end or both ends of functional features.

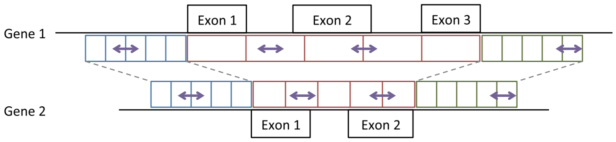

Fig. 1. To compute tag density, the upstream, body and downstream of a long (top) and short (bottom) gene are split into five blocks colored blue, red and green, respectively. While a read contributes equally in the upstream and downstream, its contribution in the body depends on block sizes

sequencing, it is essential for biologists to have access to software that can generate these plots easily.

2 Methods

2.1 Computation of tag density

iTagPlot uses an annotated list of genomic features in the BED format coupled with BED or BAM files of mapped reads to generate a tag density plot of the given feature with flanking upstream and downstream regions, the length of which is predetermined by the user (Fig. 1). Because the length of upstream and downstream is the same across all features, each block for upstream and downstream is equally divided, while the length of blocks in the body is variable due to the different lengths of features. If a fragment size is specified as in ChIP-seq, reads are lengthened to reflect the original size of fragments. To efficiently find overlap between a fragment and a block of features, all blocks are binned based on the location in the genome. To compute tag density of a block, the sum of the lengths of fragments to cover the block is divided by the length of the block, which represents the average fragment coverage for a block. To take into account different numbers of sequence reads across samples, the tag density is normalized by a specified factor: 1 000 000 as the default. A normalized tag density (NTD) for a block i is defined as $NTD_i = N\sum_j o_{ij}/(l_iT)$ where N is a normalization factor, T is the total number of reads mapped, l_i is the length of a block and o_{ij} is the overlapping length of fragment j to block i. iTagPlot allows computation of tag density around either the $5'$ or $3'$ end of features using the absolute length of blocks.

2.2 Interactive visualization

iTagPlot visualizes tag density across a functional feature such as genes, CpG islands and DNase clusters (Fig. 2). It also draws the average tag densities for a group of features or all features in a sample. A group is defined in a file or by using quantitative values for gene expression, DNA methylation and other characteristics specified for each feature. iTagPlot has a unique function to group features based on quantiles of those quantitative values and by k-means clustering that allows grouping in an unsupervised manner. Furthermore, users can use the Molecular Signatures Database (MSigDB), a collection of annotated gene groups in seven categories. Since different databases use different identifier conventions, e.g. HGNC symbol, NCBI accession and Ensembl ID, iTagPlot provides an interface to map feature identifiers in a group to those in a sample.

Supplementary Figures S1 and S2 show line and area charts for all and individual, respectively, RefSeq genes in samples. Users can zoom in and out, add text boxes, draw lines and customize drawing properties for chart type, point symbol, line width, color, font family and size, grid line, legend, title, the scale and label of X and Y and tick label, mark, width and length (Supplementary Fig. S3). Tag density values can be visualized in log scale, with smoothing and only in the visible range. iTagPlot is able to export the table of tag density values to a tab-delimited file, and chart to a PNG, EPS or PDF file for publication purposes (Supplementary Figs S3 and S4).

Since a chart is not efficient for many lines or areas (Supplementary Fig. S4 top), iTagPlot employs a heatmap viewer to efficiently visualize tag density as colors with a dendrogram, which shows how the tag density of features, groups, or samples is clustered (Fig. 2 bottom and Supplementary Fig. S4, bottom). Users can choose either hierarchical or k-means clustering, and change linkage and distance metrics. In addition to separately clustering tag density of features, groups and samples, iTagPlot also allows clustering combined tag density of chosen groups for each sample (sample-wise clustering) and that of chosen samples for each group (group-wise clustering) before clustering.

2.3 Implementation

iTagPlot was developed in Perl and Java FX. Since the number of mapped reads is usually very large, iTagPlot reduces memory requirements by assuming that mapped reads are sorted based on the starting position for each reference (See experimental results). A batch script was implemented to work with a configuration file for many features and to support parallel computation of numerous samples using multicores and a grid engine. For annotation, configuration files specify name, file path, column numbers of identifiers, numbers of blocks for body and up/downstream and length of

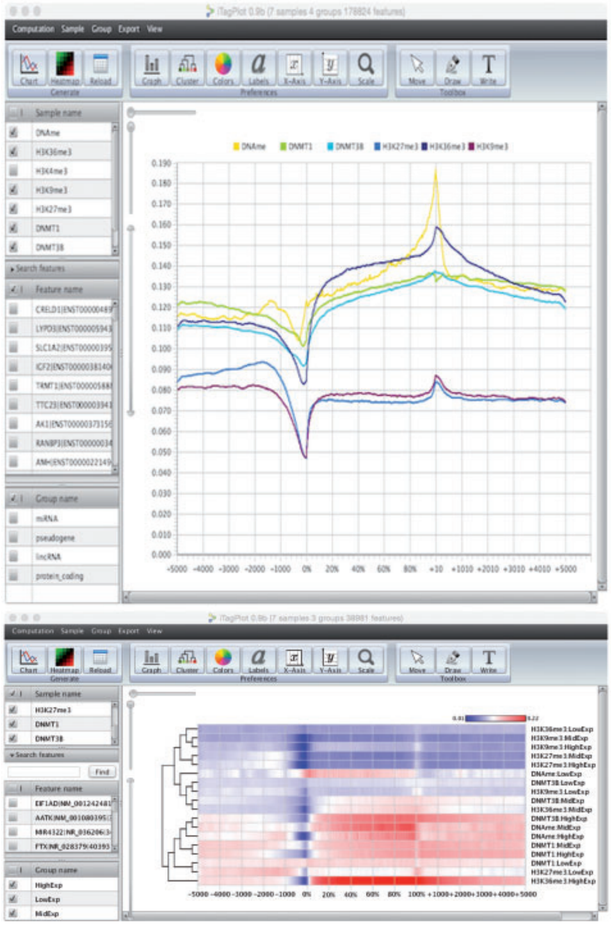

Fig. 2. Screenshot of iTagPlot line chart (top) and heatmap (bottom) for tag density across RefSeq genes

upstream and downstream. Since a tab-delimited text file to store tag density of all features for a sample could be large, the loading function of iTagPlot generates a memory index data structure and computes the average tag density for groups and samples. The drawing function reads input files again to load the tag density of chosen features. The Weka library was used for hierarchical and k-means clustering.

3 Experimental results

3.1 Materials

To demonstrate the performance of iTagPlot, we used ChIP-seq and MBD-seq datasets from differentiated NCCIT cell lines deposited on NCBI GEO (GSE38938): DNAme, DNMT1, DNMT3B, H3K27me3 and H3K4me3 from GSM952551, GSM952455, GSM952457, GSM952460 and GSM952458, respectively (Jin *et al.*, 2012). We also used Infinium 450 K bead array datasets for DNMTs and TETs knockout and control samples for the same cell line (GSE54840 and unpublished). We collected various features for the human genome from the UCSC genome browser: noncoding RNA, CpG islands, DNase clusters, RefSeq, Ensembl and GENCODE (Supplementary Table S1).

3.2 Simulated datasets

As proof of concept, we generated simulated datasets, each having a read in every 1, 10, 50, 150 and 300 bases in chromosome 21 of the human genome. In other words, all bases except both ends have the same number of reads in a dataset, but a different number of reads across datasets. Indeed, tag density plots show a plain line for this data, meaning that the computation is correct. As shown in Supplementary Figure S5, the computation is accurate regardless of the gene length: MIR3687 (60 bp), CBR1 (3,177 bp), PTTG1IP (24,318 bp) and DSCAM (834,696 bp).

3.3 Capture-based sequencing datasets

In Supplementary Figure S1, each sequencing dataset shows a distinct pattern of enrichment across RefSeq genes. As reported in many studies, H3K4me3 was highly enriched around the transcription start site (TSS), i.e. 0% in figures. While Supplementary Figure S1 shows the average tag density of all RefSeq genes, Supplementary Figure 2 shows that the tag density of genes ADORA3, DNMT1, DNMT3A and UBE2L3 for the H3K4me3 mark is quite distinct, and the last plot shows the function of highlighting a line or area and changing colors and line width.

One of the most important functions is to stratify features based on predefined groups or criteria with quantitative values such as gene expression and DNA methylation. Supplementary Figure S3 stratified RefSeq genes based on the CpG density of the promoters, i.e. high, intermediate and low CpG Promoters (HCP, ICP and LCP, respectively), and shows distinct patterns of tag density for DNA methylation and DNMT1 binding, respectively. The bottom plot was customized by adding the title, X and Y labels, and text boxes for lines and by increasing font sizes.

3.4 Score-based datasets

iTagPlot can work with score-based datasets such as beta scores from bisulfite sequencing (BS-seq) and Infinium 450 K. Supplementary Figure S6 shows methylation changes across RefSeq genes and CpG islands for Infinium 450 K with smoothing of three blocks where the Y axis represents beta scores. The DNMT1 knockdown resulted in a large amount of global demethylation relative to the control (NTC).

3.5 Running time and memory usage

Supplementary Figure S7 shows running time and memory usage for six features and seven sequencing datasets. The running time is quite variable depending on sequencing types, i.e. enrichment. Since H3K4me3 was extremely enriched in CGIs and gene promoters (Supplementary Fig. S1), this run took the longest.

Memory usage depends on the number of features because all features in a file are loaded into memory. It also depends on the maximal number of reads across a feature because those reads are loaded into memory before computing tag density. However, this is a minor factor in comparison to the number of features.

Since Infinium 450 K identified the methylation of CpGs in the same position, running time and memory usage are quite similar among datasets (Supplementary Fig. S8).

3.6 The effect of the number of blocks and the length of upstream and downstream

As shown in Supplementary Figure S9, changing the number of blocks does not affect the shape of tag density but alters the density of data points (point markers). Changing the length of upstream and downstream could generate slightly different plots because longer length causes inclusion of features not present in shorter length (Supplementary Fig. S10).

4 Conclusion

iTagPlot is a computation and interactive drawing system of tag density in a graphical user interface with many functions for customizing plots; visualizing any combination of features, groups and samples; creating groups based on a file, quantitative values, quantiles or k-means clustering; and mapping feature identifiers between groups and samples. It accurately computes tag density and was implemented to reduce memory requirements and support parallel computation as well as command line execution.

Acknowledgement

We are very grateful to anonymous reviewers for their valuable comments.

Funding

This work was supported by the National Institutes of Health [AA019976, CA185833, CA190429, and HL104125], Department of Defense [PR110715] and Brain Korea 21.

Conflict of Interest: none declared.

References

Anders,S. *et al*. (2014) HTSeq—a python framework to work with high-throughput sequencing data. *Bioinformatics*, **31**, 166–169.

Dale,R.K. *et al*. (2014) metaseq: a Python package for integrative genome-wide analysis reveals relationships between chromatin insulators and associated nuclear mrna. *Nucleic Acids Res.*, **42**, 9158–9170.

Jin,B. *et al*. (2012) Linking DNA methyltransferases to epigenetic marks and nucleosome structure genome-wide in human tumor cells. *Cell Reports,* **2**, 1411–1424.

Kent,W.J. *et al*. (2002) The human genome browser at UCSC. *Genome Res.*, **12**, 996–1006.

Liu,T. *et al*. (2011) Cistrome: an integrative platform for transcriptional regulation studies. *Genome Biol.*, **12**, R83.

Nielsen,C.B. *et al*. (2012) Spark: A navigational paradigm for genomic data exploration. *Genome Res.,* **22**, 2262–2269.

Robinson,J.T. *et al*. (2011) Integrative genomics viewer. *Nat. Biotech*., **29**, 24–26.

Shen,L. *et al*. (2014) ngs.plot: Quick mining and visualization of next-generation sequencing data by integrating genomic databases. *BMC Genomics,* **15**, 284.

Shin,H. *et al*. (2009) CEAS: cis-regulatory element annotation system. *Bioinformatics,* **25**, 2605–2606.

Statham,A. *et al*. (2010) Repitools: an R package for the analysis of enrichment-based epigenomic data. *Bioinformatics,* **26**, 1662–1663.

Ye,T. *et al*. (2011) seqMINER: an integrated ChIP-seq data interpretation platform. *Nucleic Acids Res.*, **39**, e35.