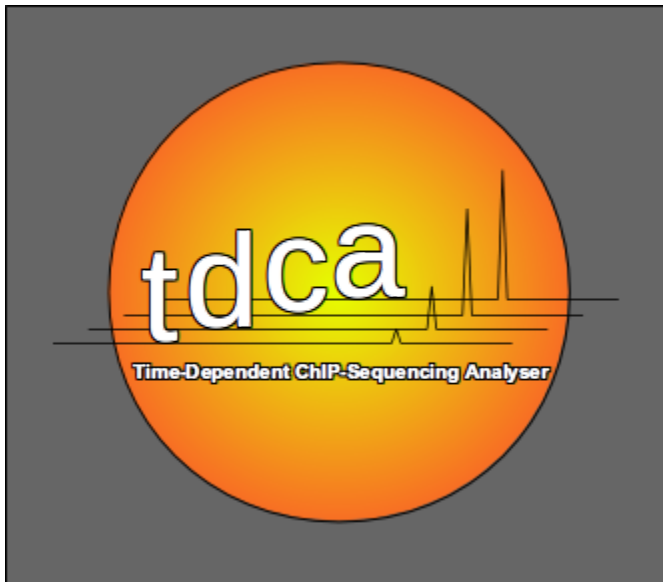
TDCA: Time-Dependent ChIP-Sequencing Analyser



Authors: Mike Myschyshyn, Marco Farren-Dai, Tien-Jui Chuang and David Vocadlo.
Download: https://github.com/luke8005/TDCA

1. Overview

1.1 Background

Chromatin immuno-precipitation followed by sequencing (ChIP-seq) is an established and robust method to generate genome wide maps of DNA binding proteins. Recently, new methods have been developed allowing time resolved ChIP-seq experiments to be conducted, effectively allowing protein-DNA binding dynamics to be established. As a response to the increasing potential of time course ChIP-seq experiments, we developed the first software specializing in time course ChIP-seq analysis. Our software, Time-Dependent ChIP-seq Analyser (TDCA), produces biologically relevant output informing users of protein-DNA binding dynamics. TDCA reads alignment data in BAM file format and genomic coordinates in BED file format.

1.3 Implementation

TDCA functionalities were developed using C++ and its graph features were built-up under R. TDCA uses samtools for bam file depth calculations and bedtools for peak intersection with genome features. TDCA makes various calls to the command line while running such as sed, awk, find, and others. Some hard coded files are created as well which users should keep in mind while using TDCA in pipelines. Samtools, bedtools, and R must be accessible from the command line.

1.4 Note about the manual

The proceeding contents of the manual contain example commands including TDCA usage. If a line starts with the $ character, it is meant to imply a command run in the terminal with the appropriate files available in the working directory. The following instructions assumes a basic understanding of terminal navigation and command execution.

2. Installation

2.1 Software Requirements

Before installing TDCA, we recommend users to install required dependencies listed below:

Name	Download Link
R	https://cran.r-project.org/bin/windows/base/ (Window)
R Package plot3D	install.packages(“plot3D”)

R Package dll	install.packages(“dll”)
R Package drc	install.packages(“drc”)
R Package rgl	install.packages(“rgl”)
R Package ggplot2	install.packages(“ggplot2”)
BedTools	https://github.com/pezmaster31/bamtools
SamTools	https://github.com/samtools/samtools

TDCA requires that bedtools (Quinlan and Hall, 2010) and samtools (Li *et al*., 2009) be installed on the user's local computer and set on the environmental variables. This is because tdca calls these programs in many of its calculations. The user can check if bedtools and samtools is accessible globally by typing “samtools” and bedtools” in the command line. If the programs are accessible, relevant information regarding the program will print. Alternatively, the user can check their bashrc file, or equivalent, with the following command:

$gedit ~/.bashrc

An alternative text editor to gedit may be used. If samtools and bedtools are set in environmental variables then the path to each of these programs directories will be documented on a line in the bashrc file like this:

export PATH=$PATH:/software/folder/bedtools/bin
export PATH=$PATH:/software/folder/samtools/bin

Although depending on how software was installed, the above lines may not be present in your bashrc file. Again, the simple way to check is typing by typing “samtools” and bedtools” in the command line.

TDCA uses the R packages drc (Ritz,C. *et al*. 2015) for curve fitting and ggplot2 for graphical output (H. Wickham, 2009). Installation of R packages can be conducted as follows:

$R
Install package:
>install.packages(“package_name”)
Check if package is installed:
>library (“package_name”)

To our knowledge, the R package drc which is used for curve fitting requires R version 3.3.1 or later.

2.1 TDCA Installation Guide on Linux

Once all the dependencies are installedSoftware Installation: Download tar file from: https://github.com/luke8005/TDCA

Unpack and navigate to the tdca directory. Assuming the unpacked tdca folder is in the home directory type:
$cd home/tdca
Run make:
$make
Now add tdca directory to environmental variables to allow accession from any directory:
$gedit ~/.bashrc
Write this line: export PATH=$PATH:home/tdca

Once tdca is added to the environmental variables, the program can be used from any directory like so:
$tdca <options>

If tdca is not added to the environmental variables, the full program path must be specified from the working directory. Ex:
$/home/tdca/tdca <options>

2.2 TDCA Installation Guide on Windows

2.2.1 Virtual Box

1. Download Oracle VM VirtualBox
2. Download Ubuntu Desktop
3. Install VirtualBox
4. In VirtualBox Manager, click New
5. Give a name to operating system and select Linux for Type and Ubuntu 32/64 bits depending on the Window OS
6. Fo RAM Memory size, give above 4GB (4096MB)
7. For Hard Disk, select Create a virtual hard disk now
8. For Hard Disk File Type, select VDI
9. Select Dynamically allocated
10. For File Location and Storage, give above 32 GB
11. In Storage, click on [Optical Drive] and select Choose a disk image
12. Open the ubuntu-version-desktop-amd32/64.iso
13. Click Start to initiate Ubuntu system in VirtualBox
14. Follow the steps to complete the Ubuntu installation

15. Once Ubuntu is successfully installed, run the command prompt and install all the necessary softwares for TDCA. Please read the installation guide on Linux in 2.1

2.3 Using TDCA

Using “-h” option in TDCA to display a list of all command line options.

Command	Description
-v	Display program version and Exit program
-h	Display a list of all command line options and Exit program
-bam	User specified folder containing sorted bam turnover files including index files.
-bed	User specified bed file containing loci of interest.
-i	User specified folder containing sorted input bam turnover files including index files.
-g	Genome name. Currently supported: mm10, mm9, hg38, hg19, dm6, dm3, ce11.
-3d	User specified gene file containing RefSeq gene names.
-s	Saturation threshold (allowable range from 0.5-0.95).
-n	User specified name for output files. DEFAULT: turnover.exp

3. General Usage Information

3.1 Supported File Format

3.1.1 BED File Format

The required BED file should contains only three columns. The format for -bed <bed_peaks.BED> is the following:

1. Chromosome, the name of the chromosome
    - Any string. ex. “Chr10”
    - Mandatory column
2. Start, the starting point
    - Any number. ex. “23507998”
    - Mandatory column
3. End, the ending point

- Any number that is greater than starting point in above. ex. “23508239”
- Mandatory column

3.1.2 BAM File Format

The BAM folder for the -bam <bam_files_folder> flag requires bam files to be sorted and indexed and named with a “XXX_integer.bam” extension, where integer is the time in minutes of the time course experiment.

3.1.3 Text File Format

The required text file for -3d <text_file> is a list of refSeq gene names separated by newlines (/n).

4. TDCA Suite

4.1 TDCA

4.1.1 Usage and Options Information

Usage: $ tdca -bed <bed_peaks.BED> -bam <bam_files_folder>

Example: $tdca -bed ChIP-seq.peaks.bed -bam bamFolder/ -i bamInputFolder/ -g mm9 -3d gene_list.txt -n exp-name

Options	**Description**
-bed <bed_peaks.BED> -bam <bam_files_folder> (Mandatory)	BED file followed by BAM files folder. Each peak in bed_peaks.BED is searched in each BAM file inside bam_files_folder in order to calculate the turnover rates.
-i <input_bam_files_folder> (Optional)	Input BAM files folder path. Input BAM files are used to normalize data.
-g <genome_name> (Optional)	Generate an additional boxplot and a heat map in the output file. The histogram displays the turnover rates in the user-given genome of interests. The heat map demonstrates turnover rates on every region in each chromosome.
-g <genome_name> -3d <text_file> (Optional)	Text file. The -3d flag requires the -g flag. A series of compressed 3D scatter plot of depth for each gene listed in the given text file is generated as PDF.
-s <0.9>	Saturation threshold (allowable range from 0.5-0.95).

(Optional)	Default is 0.9.
-n, --name <exp_name> (Optional)	Rename the output file as user-specific.

4.1.2 Default Behavior

Using only the mandatory parameters -bed and -bam, TDCA generates output containing three quality charts: peak characteristics pie chart, residual boxplot, and min/max bar chart. The analysis output provided is a distribution plot of inflection points and a scatter plot of inflection points and upper asymptotes of all peaks. TDCA was tested on H3.3 ChIP-seq data in mouse (Krauschaarr *et al.*, 2013) using eleven time points with two replicates, including input, and ~77000 peaks.

The requirement of BAM file format is specified in 3.1.1 BAM File Format. BED file format is described in 3.1.2 BED File Format.

For example (-bed <bed_peaks.BED> -bam <bam_files_folder>):

$tdca -bed ChIP-seq.peaks.bed -bam bamFolder/

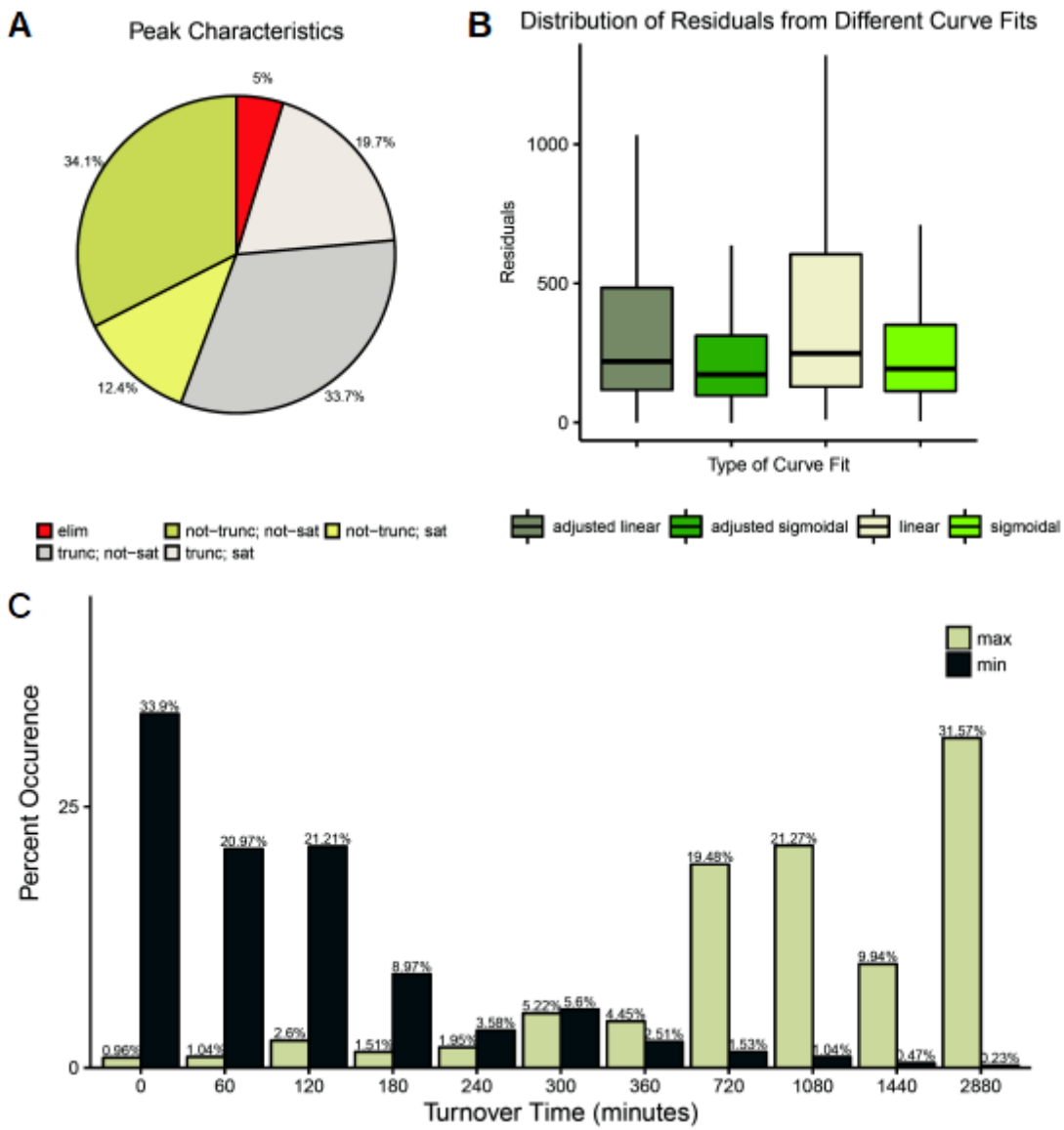


Figure 1: TDCA basic data quality output. (**A**) Peak characteristics of H3.3 turnover in mouse (Krauschaarr *et al.*, 2013). (**B**) Residuals of peaks based on different curve fits. (**C**) Bar chart of absolute minimum and maximum depth values across the different time points.

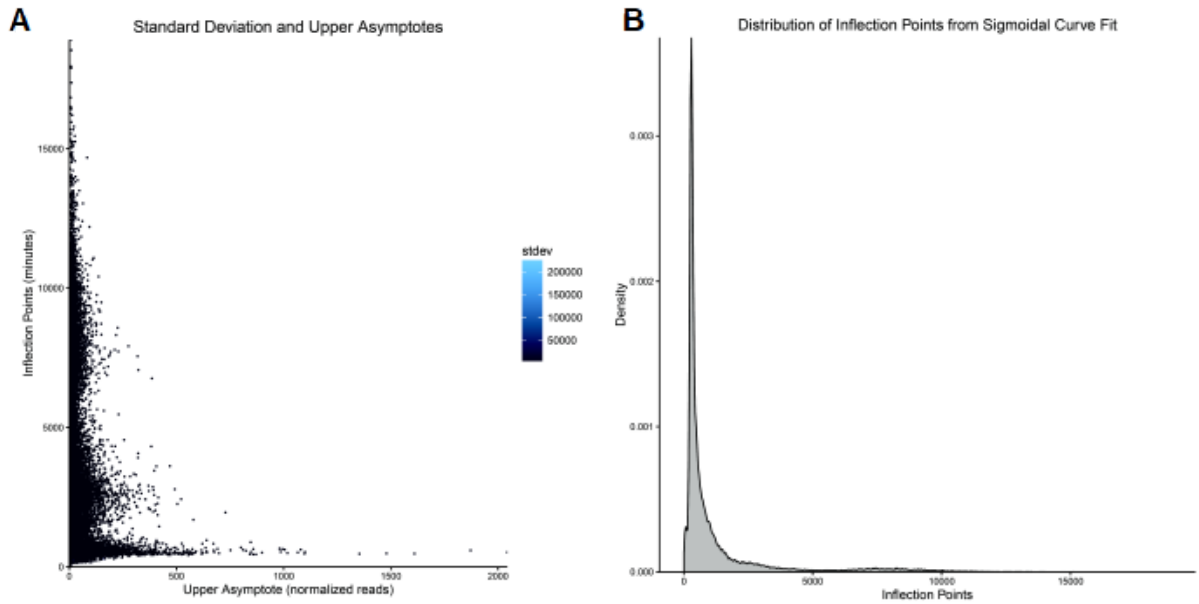


Figure 2: TDCA basic data analysis output. (**A**) Correlation of inflection points and upper asymptotes of peaks. (**B**) Distribution of inflection points.

4.1.3 Reporting the Turnover Rates with Multiple Replicate BAM Files

With additional replicated BAM Files given by the user, TDCA will take extra time to compute the analysis and graphs generation. TDCA generates an extra page of graphs for user to perform data quality comparison

The requirement of BAM file format is specified in 3.1.1 BAM File Format. BED file format is described in 3.1.2 BED File Format.

For example (-bed <bed_peaks.BED> -bam <bam_files_folder> -bam <replicated_bam_file_folder_1>):

$tdca -bed ChIP-seq.peaks.bed -bam rep1-bamFolder/ -bam rep2-bamFolder/

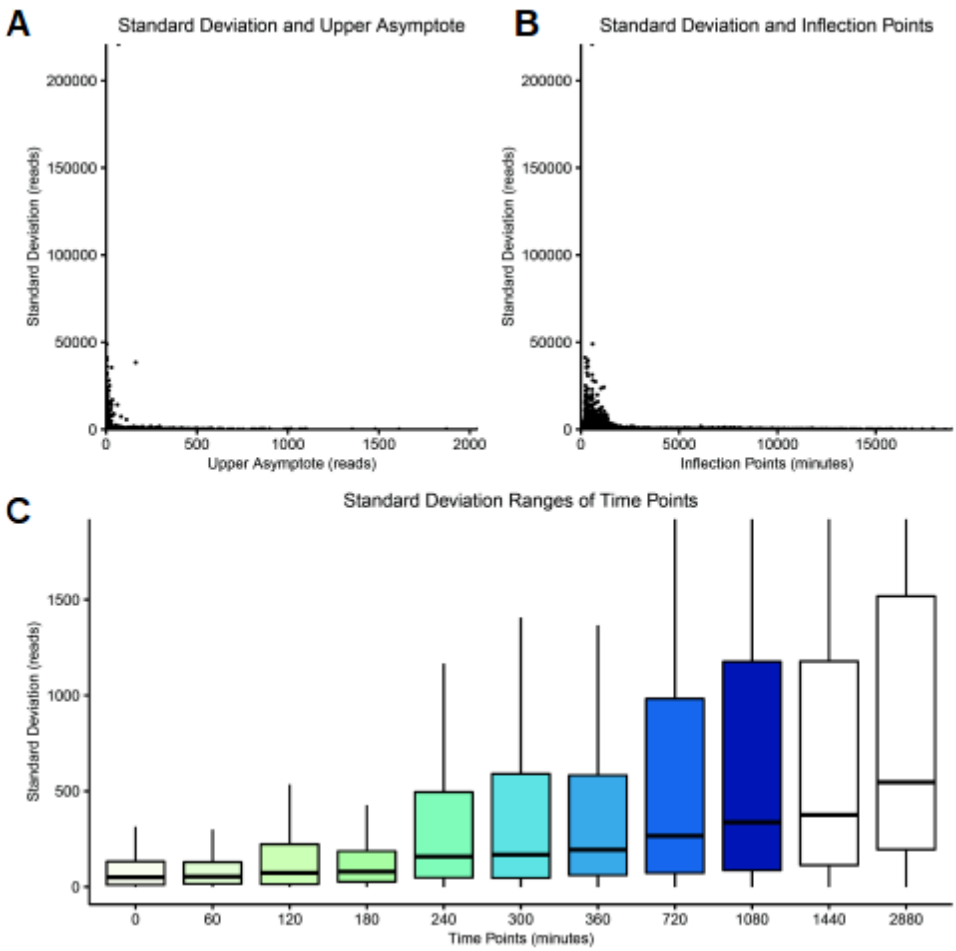


Figure 3: TDCA output with replicates. (**A**) Correlation of standard deviation and upper asymptotes of peaks. (**B**) Correlation of standard deviation and inflection points of peaks. (**C**) Distribution of standard deviation at different time points.

4.1.4 Reporting the Turnover Rates of Given BAM Files with Inputs (-i)

Given BAM input files from the user, additional normalization via subtracting depth of input to a lower limit of zero will be computed.

The requirement of BAM file format is specified in 3.1.1 BAM File Format. Example:

$tdca -bed ChIP-seq.peaks.bed -bam rep1-bamFolder/ -i rep1-bamInputFolder/ -bam rep2-bamFolder/ -i rep2-bamInputFolder/

4.1.5 Reporting the Turnover Rates of Given BAM Files with the User-Specific Genome Inside Built-In Genome Library (-g)

Given a user-specified genome that is supported, TDCA generates a box plot of infection points in minutes for genomic features. Default genome features include:

3’UTR exon and 1000bp upstream of transcriptional start site, 5’UTR exon and 1000bp downstream of transcriptional end site, coding exons, CpG islands, introns, whole gene - characterized as 1000bp upstream of transcriptional start site to 1000bp downstream of transcriptional end site, and intergenic region - characterized as reciprocal coordinates of whole gene. Additional gene features can be included by the user in a bed file format. This process is described in section 4.1.8 Expanding Genome Feature Libraries.

In addition, TDCA generates an ideogram heatmap of turnover infection points at each canonical chromosome. The strength of the density is labeled by different colours

For Example (-bed <bed_peaks.BED> -bam <bam_files_folder> -g <genome_name>):

$tdca -bed ChIP-seq.peaks.bed -bam bamFolder/ -i bamInputFolder/ -g mm9

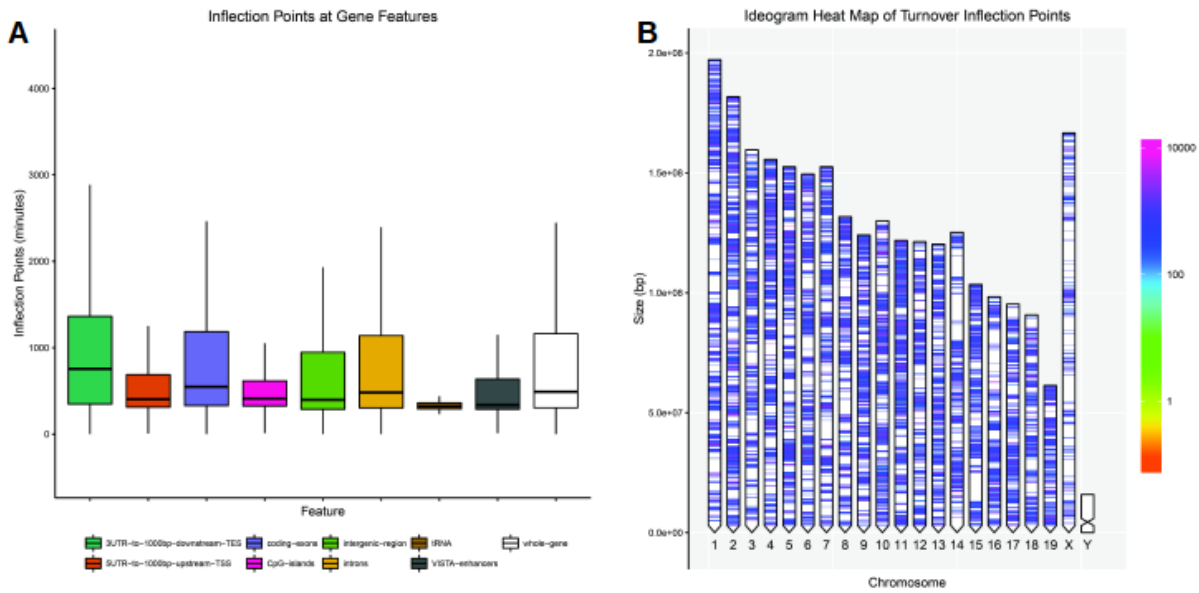


Figure 4: TDCA output with specified genome. (**A**) Distribution of inflection points at different gene features. (**B**) Ideogram heatmap of inflection points across conical chromosomes.

Supported genomes include human (hg19, hg38), mouse (mm9, mm10), fly (dm3, dm6), and nematode (ce11). Contact the authors to request additional genomes.

4.1.6 3D depth scatter plots of genes (-g -3d)

An additional feature when a user-specified genome is provided can be provided as a series of 3D scatter plots displaying time dependent depth for select

genes from a user specified gene list. The user-specified gene list is a text file generated by the user with refSeq gene names separated by newline characters (\n).

The required format of the user-specified gene list is specified in 3.1.3 Text File Format.

For Example (-bed <bed_peaks.BED> -bam <bam_files_folder> -g <genome_name> -3d <text_file>):

$tdca -bed ChIP-seq.peaks.bed -bam bamFolder/ -i bamInputFolder/ -g mm9 -3d chr.txt

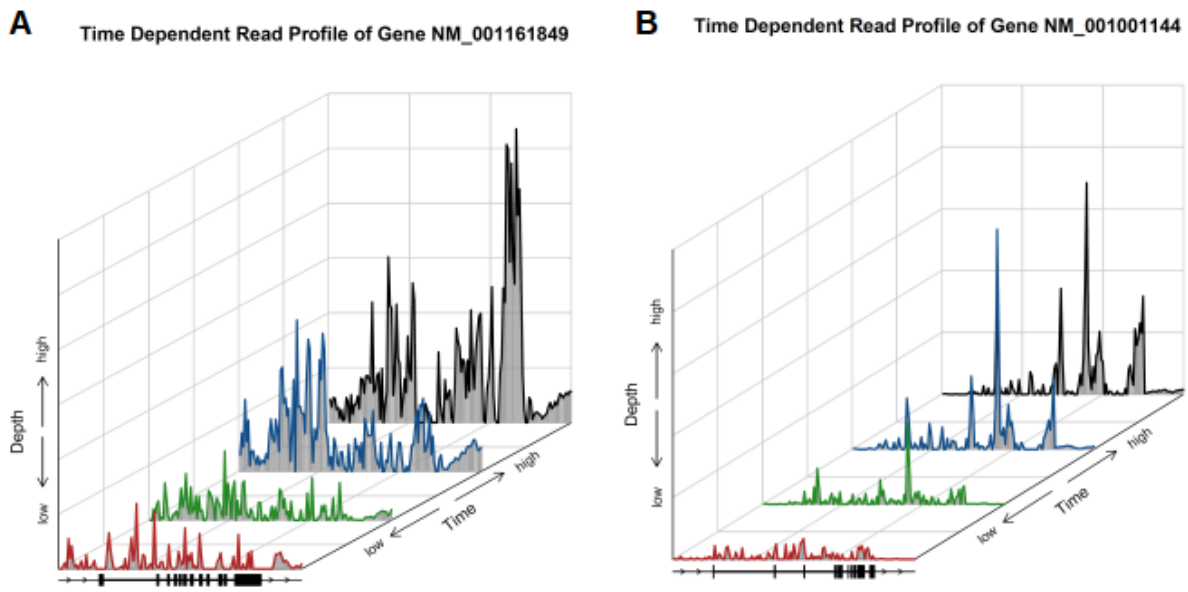


Figure 5: 3D depth scatter plots of NM_001161849 (**A**) and NM_001001144 (**B**).

The 3D scatter plot option shows time on the z axis (into the page). Data is compressed to show four time bins. This is done so that the plot is not cluttered. The compression processes is visually described in the figure 6. The purpose of this diagram is to get a relative idea of turnover times for various peaks located at genes.

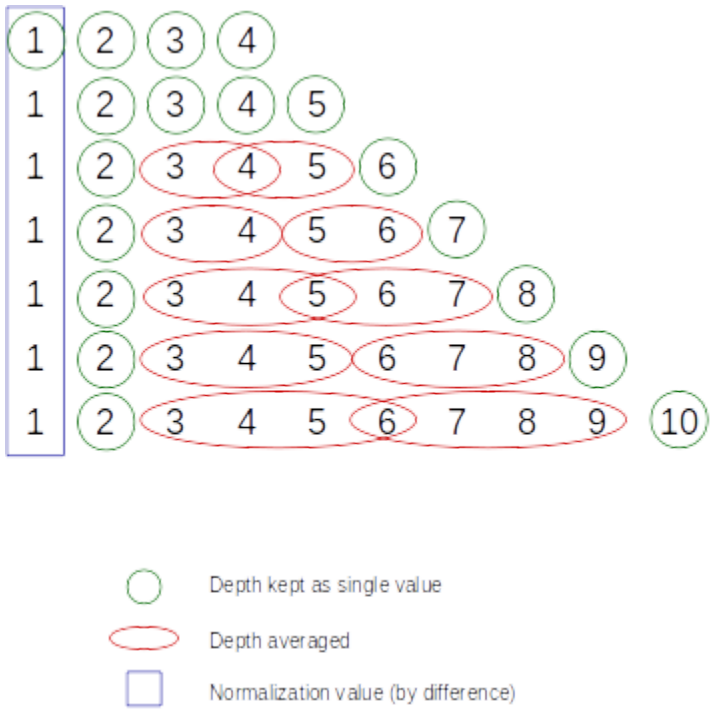


Figure 6: Visual display of 3D scatter plot algorithm. Horizontal lines containing numbers represent the amount of time point in a given ChIP-seq time course experiment. The amount of time points shown in the scatter plot is set to four no matter how many additional time points are given as can be seen from the green or red circled time points. The first time point is used as a normalization point by subtracting its depth from the other data points. An experiment with only four time points would show all four each normalized by the first, therefore the first time point will look like a flat line.

The x axis of all 3D genes show a picture of the exons in black boxes and introns in thicker black lines. 1000bp upstream and downstream regions of the gene are shown as thinner black lines on the left and right sides of the gene body respectively. An algorithm has been created to search a build in library of refSeq gene information. Keep in mind that refSeq genes have multiple isoforms of certain genes. Remember to choose the appropriate isoform. USCS browser is a great tool to visually inspect coordinates of isoforms (Kent,W.J. *et al*. 2002).

4.1.7 Reporting Turnover Rates with Different Saturation Threshold

TDCA offers a saturation threshold flag (-s). By default, saturation is defined as at least one data point closest to the absolute maximum for forward turnover to be within 90% of its value or absolute minimum for reverse turnover to be within 110%

of its value. This threshold can be adjusted by the user to be between 0.55-0.95. Figure 7 shows the difference in peaks considered saturated using two different -s values.

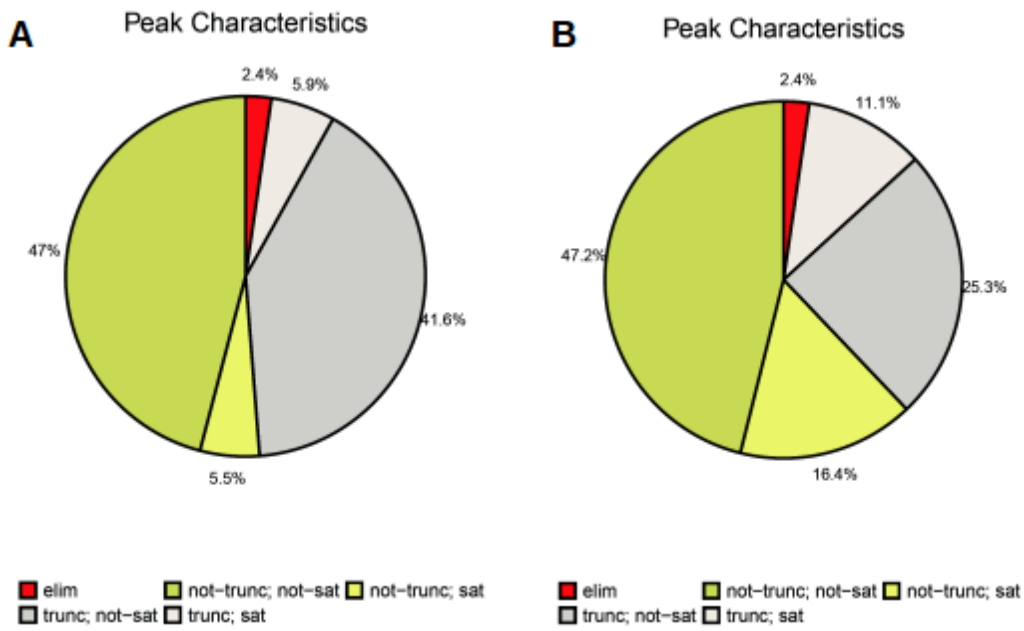


Figure 7: Peaks characteristics with default (0.9) saturation threshold (**A**) and 0.75 saturation threshold (**B**).

4.1.8 Expanding Genome Feature Libraries

Users can input their own BED file format genome feature into the appropriate genome folder located in the TDCA GenomeFeatures folder. UCSC table browser was used to get default libraries (Karolchik D., *et al*. 2004). TDCA will use the newly input file in analysis in of inflection points at genome features, as shown in figure 8.

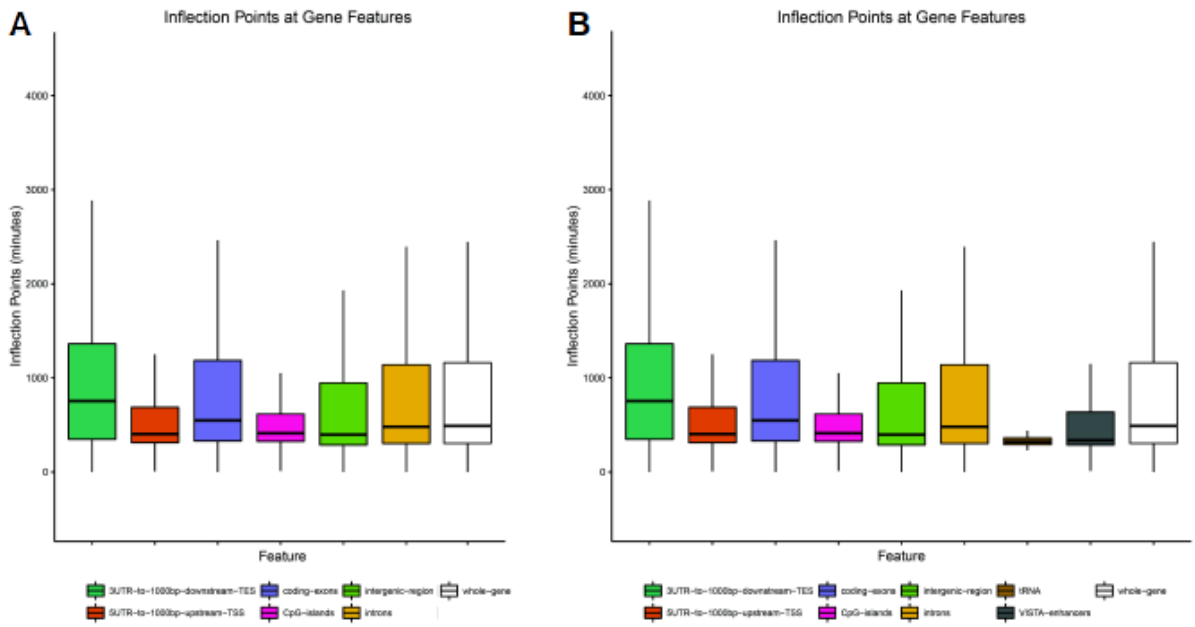


Figure 8: Inflection points at default genome features (**A**) and genome features with tRNA genes and enhancer coordinates added (**B**).

5. Example Usage

5.1 Getting data

During development, the example we used to test our software is data from a H3.3 ChIP-seq time course study. The accession number for project is GSE51505 (found at: http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE51505)

Accession numbers and names of data we will use for example:

GSM1246648	MEF_H3.3_0h_r1
GSM1246649	MEF_H3.3_1h_r1
GSM1246650	MEF_H3.3_2h_r1
GSM1246651	MEF_H3.3_3h_r1
GSM1246652	MEF_H3.3_4h_r1
GSM1246653	MEF_H3.3_5h_r1
GSM1246654	MEF_H3.3_6h_r1
GSM1246655	MEF_H3.3_12h_r1
GSM1246656	MEF_H3.3_18h_r1
GSM1246657	MEF_H3.3_24h_r1
GSM1246658	MEF_H3.3_48h_r1
GSM1246659	MEF_H3.3_72h_r1
GSM1246660	MEF_H3.3_0h_r2

GSM1246661	MEF_H3.3_1h_r2
GSM1246662	MEF_H3.3_2h_r2
GSM1246663	MEF_H3.3_3h_r2
GSM1246664	MEF_H3.3_4h_r2
GSM1246665	MEF_H3.3_5h_r2
GSM1246666	MEF_H3.3_6h_r2
GSM1246667	MEF_H3.3_12h_r2
GSM1246668	MEF_H3.3_18h_r2
GSM1246669	MEF_H3.3_24h_r2
GSM1246670	MEF_H3.3_48h_r2
GSM1246671	MEF_H3.3_0h_Input
GSM1246672	MEF_H3.3_1h_Input
GSM1246673	MEF_H3.3_2h_Input
GSM1246674	MEF_H3.3_3h_Input
GSM1246675	MEF_H3.3_4h_Input
GSM1246676	MEF_H3.3_5h_Input
GSM1246677	MEF_H3.3_6h_Input
GSM1246678	MEF_H3.3_12h_Input
GSM1246679	MEF_H3.3_18h_Input
GSM1246680	MEF_H3.3_24h_Input
GSM1246681	MEF_H3.3_48h_Input
GSM1246682	MEF_H3.3_72h_Input

Download the above SRA experiments and unpack using sra toolkit (Leinonen,R. *et al*. 2011) fastq-dump command. Data then needs to be aligned (Li H and Durbin R. 2009) to the mouse genome and peaks need to be called (Zhang,Y. *et al*. 2008). We used the following commands for this:

Align to mm9:
$bwa mem mm9.fa MEF_H3.3_0h_r1 > MEF_H3.3_72h_r1.mm9.mem_0.sam
Convert sam to bam:
$samtools view -bS MEF_H3.3_72h_r1.mm9.mem_0.sam >
MEF_H3.3_72h_r1.mm9.mem_0.bam
Sort bam:
$samtools sort MEF_H3.3_72h_r1.mm9.mem_0.bam
MEF_H3.3_72h_r1.mm9.mem.sorted_0.bam
Remove duplicates if any:
$samtools rmdup MEF_H3.3_72h_r1.mm9.mem.sorted_0.bam
MEF_H3.3_72h_r1.mm9.mem.sorted.rmdup_0.bam
Create bam index:
$samtools index MEF_H3.3_72h_r1.mm9.mem.sorted.rmdup_0.bam
MEF_H3.3_72h_r1.mm9.mem.sorted.rmdup_0.bam.bai

Repeat this for each fastq file. Note that the bam files are named with the convention “XXX_integer.bam”. This is naming convention is essential for TDCA to detect the time point in question. TDCA uses regex to do this and currently support only integer times in minutes. Call peaks using time point with longest treatment time - 4320 minutes (72 hours of doxycycline treatment).

Call broad peaks with macs2 (77531 peaks):

$macs2 callpeak -t MEF_H3.3_72h_r1.mm9.mem.sorted.rmdup_4320.bam -c MEF_H3.3_72h_input.mm9.mem.sorted.rmdup_4320.bam --broad -g mm -n h3.3.72h-72hinput.macs2-broad.0.05 --broad-cutoff 0.05

The bed file required for TDCA input must contain 3 tab delimited columns: chromosome, start, and end for each peak. Copy and paste these into a text file from the macs2 xls output.

Once data is obtained, put all the bam files and indices with correct name extension for time points (XXX_integer.bam) in a folder for each replicate and input.

For example:
Make a directory for replicate 1 files:
$mkdir krauschaarr-rep1
Move files replicate 1 files to newly created directory:
$mv -t ./krauschaarr-rep1 MEF_H3.3_48h_r1.mm9.mem.sorted.rmdup_2880.bam
MEF_H3.3_72h_r1.mm9.mem.sorted.rmdup.bam
MEF_H3.3_0h_r1.mm9.mem.sorted.rmdup_0.bam
MEF_H3.3_1h_r1.mm9.mem.sorted.rmdup_60.bam
MEF_H3.3_3h_r1.mm9.mem.sorted.rmdup_180.bam
MEF_H3.3_5h_r1.mm9.mem.sorted.rmdup_300.bam
MEF_H3.3_12h_r1.mm9.mem.sorted.rmdup_720.bam
MEF_H3.3_24h_r1.mm9.mem.sorted.rmdup_1440.bam
MEF_H3.3_48h_r1.mm9.mem.sorted.rmdup_2880.bam
MEF_H3.3_6h_r1.mm9.mem.sorted.rmdup_360.bam
MEF_H3.3_18h_r1.mm9.mem.sorted.rmdup_1080.bam
MEF_H3.3_2h_r1.mm9.mem.sorted.rmdup_120.bam
MEF_H3.3_4h_r1.mm9.mem.sorted.rmdup_240.bam
MEF_H3.3_72h_r1.mm9.mem.sorted.rmdup.bam
MEF_H3.3_0h_r1.mm9.mem.sorted.rmdup_0.bam.bai
MEF_H3.3_24h_r1.mm9.mem.sorted.rmdup_1440.bam.bai
MEF_H3.3_4h_r1.mm9.mem.sorted.rmdup_240.bam.bai
MEF_H3.3_12h_r1.mm9.mem.sorted.rmdup_720.bam.bai

MEF_H3.3_2h_r1.mm9.mem.sorted.rmdup_120.bam.bai
MEF_H3.3_5h_r1.mm9.mem.sorted.rmdup_300.bam.bai
MEF_H3.3_18h_r1.mm9.mem.sorted.rmdup_1080.bam.bai
MEF_H3.3_3h_r1.mm9.mem.sorted.rmdup_180.bam.bai
MEF_H3.3_6h_r1.mm9.mem.sorted.rmdup_360.bam.bai
MEF_H3.3_1h_r1.mm9.mem.sorted.rmdup_60.bam.bai
MEF_H3.3_48h_r1.mm9.mem.sorted.rmdup_2880.bam.bai
MEF_H3.3_72h_r1.mm9.mem.sorted.rmdup_4320.bam.bai

Repeat this for both replicates and input. If the user is working with the two replicates and input from Krauschaarr et al. (2013) then there should be three folders corresponding to the two replicates and and the input, each containing appropriately named am files and indices. The working directory should also contain a bed file of H3.3 peaks.

$ls
krauschaarr-rep1
krauschaarr-rep2
krauschaarr-input
H3.3.72h-72hinput.macs2-broad.0.05.chr10.bed

Note that the H3.3 data only has time point 4320 for replicate 1. TDCA will give an error if replicates have differing timepoints. So the name of the bam file for time point 4320 was changed from MEF_H3.3_72h_r1.mm9.mem.sorted.rmdup_4320.bam to MEF_H3.3_72h_r1.mm9.mem.sorted.rmdup.bam. The absence of the “XXX_integer naming convention will make it invisible to TDCA.

Alternatively, we provide pre-aligned chromosome 10 data along with peaks for faster testing. This data can be found here:
https://drive.google.com/open?id=0B5BFPUpdPrmhdG5jbkFUSlEtZDA

5.2 Running TDCA

These files now satisfy the basic input requirement to run TDCA. TDCA commands could be ran as follows:

Run replicate 1 with no genome specified (heatmap ideogram and gene features boxplot will not be created):
$tdca -bed H3.3.72h-72hinput.macs2-broad.0.05.chr10.bed -bam krauschaarr-chr10-rep1/ -n r1.chr10.minimum

Run both replicates and input and specify genome:
$tdca -bed H3.3.72h-72hinput.macs2-broad.0.05.chr10.bed -bam krauschaarr-chr10-rep1/ -bam krauschaarr-chr10-rep2/ -i krauschaarr-chr10-input/ -i krauschaarr-chr10-input/ -g mm9 -n r1.r2.input.chr10.mm9

6. FAQ

6.1 Installation fails for R package, “rgl” in Ubuntu environment because of X11 not found but required.

Run the following in command line, sudo apt-get install r-cran-rgl.

6.2 Generate pdf file without compiling tdca.

Once the R scripts are generated by tdca, run the following in command line, Rscript name_R_script. “xxx.tdca3Dgenes.R” is used to generate 3D graphs and “xxx.tdca.R” generates default graphs

6.3 Changing the look of output graphs.

R scripts are provided for each graphical output. Users may wish to change the look of certain graphs and can do so with a basic understanding or R and ggplot2. R scripts are generated in a modular fashion to facilitate single change options. Data generated from larger genomes may create extremely large R scripts and may not open well with all text editors. The format of all R scripts is the same given the same tdca flag calls. Thus R scripts can also be manipulated by line swapping using combinations of awk and cat or other commands.

6.4 What compiler do I need to install tdca?

TDCA is compiled using g++. Most later versions should work and version 4.9.3 has been tested exhaustively. C++ standard library 2014 (-std=c++14) is used in the TDCA Makefile, however C++ standard library 2011 also works. Users may change the -std=c++14 flag to -std=c++11 in the TDCA Makefile if they wish.

7. Runtime Dependencies

7.1Number of Processors/BED File Peaks/ BAM files

TDCA is parallelized to run on all available processors. Runtime dependencies were tested using replicates from Krauschaarr et al. (2013) on the bugaboo server of westgrid computer cluster.

In the first series of runtimes, only the first replicate was used. Eleven time points were used, and these BAM files which were trimmed to only chromosome 10. Accordingly, The BED file utilized contained only peaks from this chromosome.

Runtimes were found to increase linearly with number of BED file peaks. Rate of increase was roughly an additional 3 minutes for every additional 1000 peaks in

the BED file; this rate was essentially unchanged regardless of number of processors used while running the program. There was no significant reduction in runtime for parallelization beyond 4 processors

Figure 9 shows a summary of runtimes as a function of number of peaks in the BED file run with different number of processors.

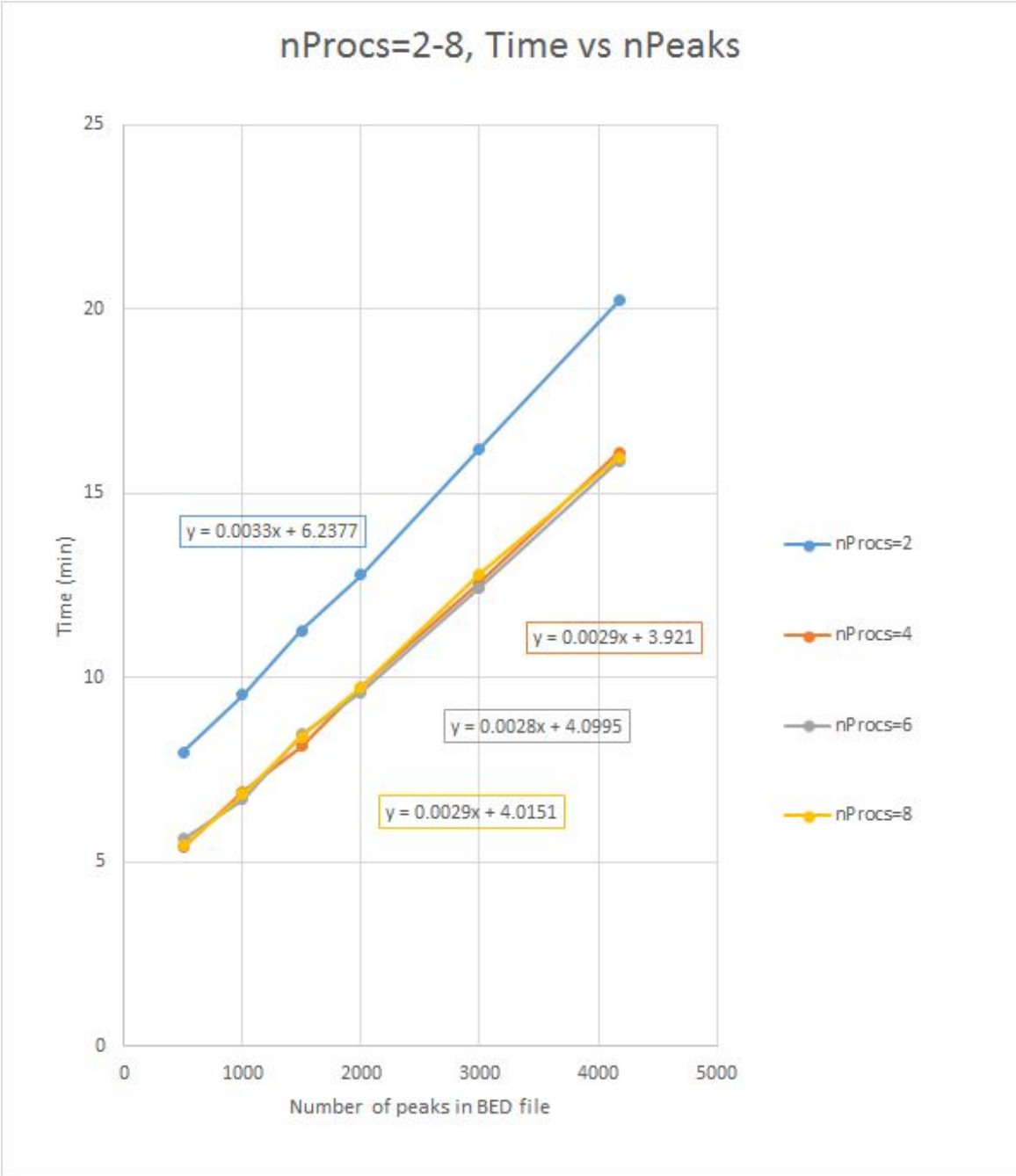


Figure 9: runtimes as a function of number of peaks in the BED file (nPeaks=500, 1000, 1500, 2000, 3000, 4180). This relationship was examined using varying number of processors (nProcs=2, 4, 6, 8).

8. TDCA Support

Please submit bug reports and request for library expansions to: mmyschyshyn@gmail.com

9. References

Karolchik D., *et al*. (2004) The UCSC Table Browser data retrieval tool. *Nucleic Acids Res.* **1**, D493-6.

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

Kraushaar,D.C. *et al*. (2013) Genome-wide incorporation dynamics reveal distinct categories of turnover for the histone variant H3.3. *Genome Biol.*, **14**, R121.

Leinonen,R. *et al*. (2011) The Sequence Read Archive. *Nucleic Acids Res*. **39**, D19-D21.

Li H and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **15**, 1754-60.

Li,H. *et al.* (2009) The Sequence alignment/map (SAM) format and SAMtools. *Bioinformatics*, **25**, 2078-9.

Wickham,H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

Quinlan,A.R. and Hall,I.M. (2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**, 841-842.

Ritz,C. *et al*. (2015) Dose-Response Analysis Using R. *PLos One*, **10**, e0146021.

Zhang,Y. *et al*. (2008) Model-based Analysis of ChIP-Seq (MACS). *Genome Biol.* **9**, R137.