DANPOS: Dynamic Analysis of Nucleosome Position and Occupancy by Sequencing

Kaifu Chen[1], Yuanxin Xi[1], Xuewen Pan[2, 3], Jessica Tyler[4, 5], Sharon Dent[5, 6], Xiangwei He[2], Wei Li[1*]

[1]Division of Biostatistics, Dan L Duncan Cancer Center and Department of Molecular and Cellular Biology, [2]Department of Molecular and Human Genetics, [3]Verna and Marrs McLean Department of Biochemistry and Molecular Biology, Baylor College of Medicine, Houston, TX 77030, USA
[4]Center for Cancer Epigenetics, [5]Department of Biochemistry and Molecular Biology, [6]Department of Molecular Carcinogenesis, Science Park, The University of Texas MD Anderson Cancer Center, Houston, TX 77030
* Corresponding Author. Telephone: 713.798.7854; email: WL1@bcm.edu

For information, please contact:

chenkaifu@gmail.com or wl1@bcm.edu

CITIATION:
Chen, K., Xi, Y., Pan, X., Li, Z., Kaestner, K., Tyler, J., et al. (2012). DANPOS: Dynamic Analysis of Nucleosome Position and Occupancy by Sequencing. *Genome Research*. doi:10.1101/gr.142067.112

THE WORK FLOW

Figure-1 (Li)

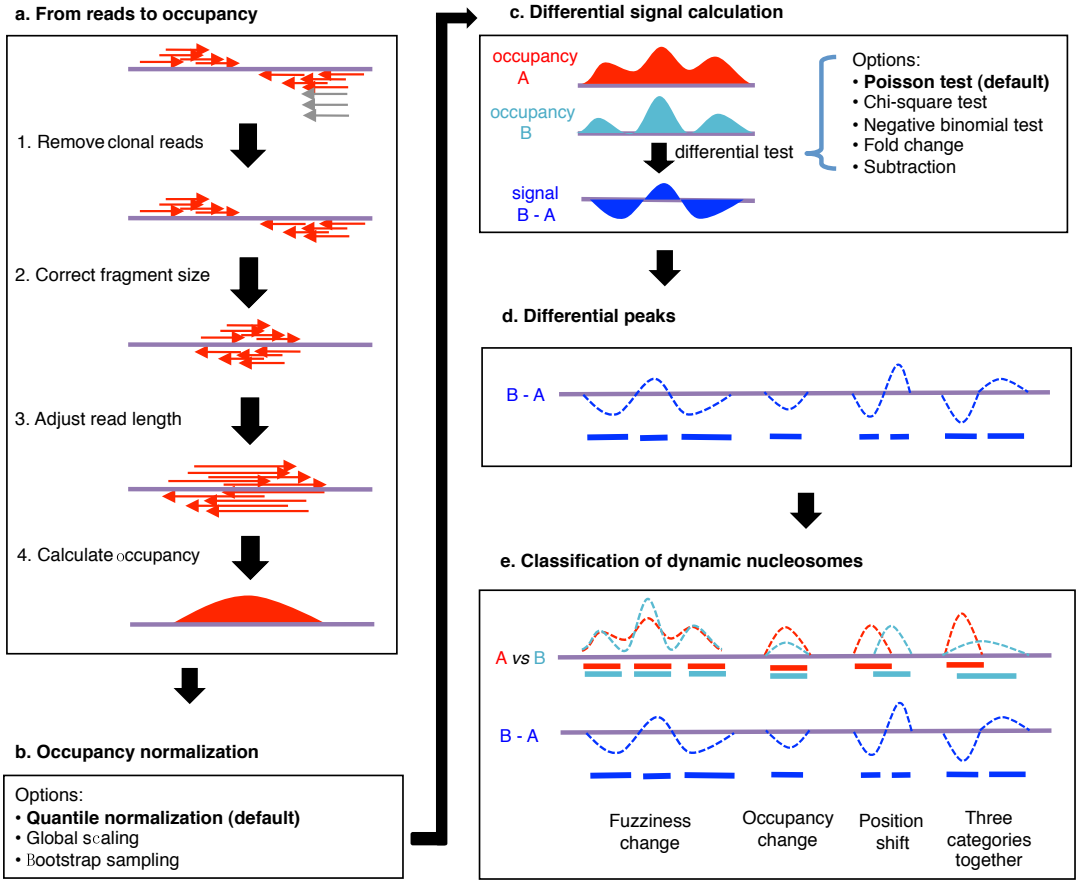

Figure 1: Flowchart of the nucleosome differential analysis pipeline in DANFOSS. (**a**) Schematic illustration of nucleosome occupancy calculations. Purple line represents the DNA sequence. Red arrows indicate normal reads at both ends of a nucleosome, whereas gray arrows show clonal reads. Red areas indicate the calculated nucleosome occupancy. (**b**) The alternative data normalization methods. (**c**) Calculation of differential signals at single nucleotide resolution. Red and sky-blue areas represent nucleosome occupancies in samples A and B, respectively. Blue area represents the differential signal at single nucleotide resolution between samples A and B. Five statistical methods that can be used to calculate differential signal are listed on the right. (**d**) Differential peaks (dynamic nucleosome). Blue dashed lines represent differential signals at single nucleotide resolution. Blue bars represent differential nucleosome peaks called based on the differential signal. (**e**) A cartoon to show the 3 categories of nucleosome changes. Red and sky-blue dashed lines represent nucleosome occupancy in samples A and B, respectively. Red and sky-blue blocks represent occupancy peaks (nucleosomes) from samples A and B, differential signals and peaks were displayed as in (d).

Starting from sequencing reads that have been mapped to the reference genome, there are 5 steps in the DANPOS workflow (Fig. 1). The first step is preliminary data processing that calculates nucleosome occupancy from the mapped reads (Fig. 1a). DANPOS provides an option to remove clonal reads with identical sequences resulting from possible over-amplification during sample preparation. We determine clonal reads based on their extremely high coverage relative to the mean coverage across the

genome based on a Poisson P value cutoff. Although nucleosome size is 147 bp in higher eukaryotes, the real size of DNA fragments after MNase digestion can vary from approximately 120 bp to 170 bp [1,2]. This variation can be compensated for by shifting each read toward the 3' direction for half of the estimated fragment size. For single-end sequencing, the average fragment size can be estimated by the distribution of distances between reads on the positive and negative strands, whereas for paired-end sequencing the fragment size is determined directly by the distance between the paired-end reads. The read length is then adjusted to half of the nucleosome size (i.e., 74 nt) to enhance the signal-to-noise ratio [3]. Finally, nucleosome occupancy is calculated as the count of adjusted reads covering each base pair in the genome.

The sequencing depths of different samples also vary in MNase-seq experiments. Therefore, to make the occupancy levels comparable, DANPOS provides several normalization options in its second step (Fig. 1b), including quantile normalization (default), global scaling, and bootstrap sampling. In the third step (Fig. 1c), DANPOS calculates the nucleosome differential signal at single-nucleotide resolution between control and treatment samples based on a Poisson test (default), which has been widely used in sequencing data analysis [4]. Alternative statistical methods are also supported, such as Pearson's Chi-square test [5], negative binomial test [6], fold-change and numerical subtraction. In the fourth step, DANPOS performs peak calling on the single-nucleotide-resolution differential signal to identify differential peaks (Fig. 1d). Finally, in the fifth step, DANPOS provides scores of fuzziness change, occupancy change, and shif distance in the result file, and thus allow user to do the classification and retrieve each categories based on self-defined cutoffs (Fig. 1e).

INSTALLATION

The package is available at: http://code.google.com/p/danpos/, simpley download the latest version to a directory and extract the file. For Linux system, extract by a command line:

tar –xvzf danpos-#.#.#.tgz

Then go to the directory danpos-#.#.# and try:

Python danpos.py

If everything is fine, you will get the following message:

usage:

python danpos.py <path> [optional arguments]

for more help, please try: python danpos.py -h

To run DANPOS on a test dataset, go to the directory danpos-#.#.#/test and try:

python ../danpos.py file1.bed:dir2,file1.bed:dir3/,dir2/:dir3

Or download and extract a test data set at:
http://dldcc-web.brc.bcm.edu/lilab/kaifuc/danposTestData/testDataGSE29292.tgz
go to the directory testDataGSE29292 and type the following command:

python path2danpos/danpos.py shut/:wild -p 1

If everything is fine, you will finally see a message:

Job done, cheers!

PREREQUISITE
* The latest version is tested on a Linux system with R version 2.12.2, Python 2.7, rpy2, and numpy 1.5.0.
* Samtools need to be installed in the system if you have input data in .bam or .sam format.
* The R package preprocessCore need to be installed in your system if you are going to use Quantile normalization.
* Memory ~ (genome_size/step_size) x (replicate_count + max(2, comparison_count)) x 8 bits.

USAGE
python danpos.py <path> [optional arguments]

positional arguments:
```
 path              Pairs of paths to MNaseSeq data sets. The two paths in
                   each pair must be seperated by ':', a:b means a minus
                   b, different pairs must be seperated by ',' e.g.
                   file1.bed:dir2/,dir3/, each path could point to a file
                   or a directory containing multiple files, files under
                   each directory represent multiple replicates for the
                   same group.
```

optional arguments:
```
 -h, --help        show this help message and exit
 -p , --paired     set to 1 if the input data is paired-end reads. Ignore
                   this when the input is wiggle format occupancy data
                   (default: 0)
 -b , --bg         pairs of paths, each pair secify a genomic background
                   data set for a MNaseSeq data set, a:b means b is the
                   background data set for a, put a word 'None' when a
                   MNaseSeq data set has no background data set, e.g.
                   file1.bed:bgdir1,dir2/:None,dir3/:bg3.bed, this
```

```
                function is useful only when the genomic background
                between groups are different, e.g. the transpons in
                the genome is significantly amplified in one group,
                for most other cases, this function is not
                recommended. (default: None)
-c , --count        specify the count of reads per replicate for each
                group, e.g.
                file1.bed:10000000,dir2/:20000000,dir3/:15000000, so
                the reads count per replicate for each group will
                finally be normalized to these count. Do this only
                when you are clear about what you are doing, e.g. when
                you have spike-ins to measure the real reads count in
                each replicate (default: None)
-o , --out          a name for the output directory (default: result)
-q , --height       occupancy/intensity cutoff for nucleosome calling
                (default: 5)
-t , --testcut      P value cutoff for calling differential nucleosomes
                (default: 1e-5)
-n , --nor          data normalization method, could be 'F','Q','S' or
                'N', representing quantile normalization, fold
                normalization, normalize by sampling, or no
                normalization (default: F)
-w , --width        minimal width of peak (default: 40)
-d , --distance     minimal distance between peaks, peaks closer than d
                will be merged as on peak (default: 100)
-e , --edge         set to 1 if need to detect edges for
                occupancy/intensity peaks,else set to 0 (default: 0)
-a , --span         the span or step size in the generated wiggle data
                (default: 10)
-k , --keep         save middle stage files? set to 0 if don't save,
                otherwise set to 1 (default: 0)
-z , --smooth_width   the smooth width before peak calling, set to 0 if need
                not to smooth (default: 20)
-x , --pcfer        set to 1 if want to do nucleosome calling for each
                replicate,else set to 0 (default: 0)
-l , --lmd          lambda width for smoothing background data before
                background subtraction, ignore this when the parameter
                -b is not specified. (default: 300)
-g , --gapfill      do gap filling? for nucleosome calling, fill gap
                between nucleosomes if the gap size is similar to
                nucleosome size, set to 0 if don't fill, otherwise set
                to 1 (default: 0)
--clonalcut         the cutoff for adjusting clonal signal, set as a P
                value larger than 0 and smaller than 1, set as 0 if
                don't need to adjust clonal signal, or set as 1 to
                allow automative detection. (default: 1)
```

--frsz specify the average size of DNA fragments in the seuqnecing experiment. By default it is automatically detected by DANPOS. Ignore this when the input is wiggle format occupancy data (default: None)

--mifrsz minimal size of the DNA fragments, DANPOS will select a most probable frsz value within the range between --mifrsz value and --mafrsz value. ignore this when ' --frsz' has been specified. Ignore this when the input is wiggle format occupancy data (default: 50)

--mafrsz maximal size of the DNA fragments, DANPOS will select a most probable frsz value within the range between --mifrsz value and --mafrsz value. ignore this when ' --frsz' has been specified. Ignore this when the input is wiggle format occupancy data (default: 250)

--extend specify the size theat each fragment will be adjusted to, the size of each fragment will be adjusted to this size when reads data is converted to occupancy data. Ignore this when the input is wiggle format occupancy data (default: 80)

INPUTS

The input can be occupancy data in standard **Wiggle fixedstep format** (the file name must be end with '.wig') or sequencing reads that has been mapped to a reference genome and stored in standard .bed, .bam, .sam format files. The default out format of bowtie is also supported, but please use ".bowtie" as the end of file name. **For wiggle format, please use a unique step size for all input files and set the DANPOS paramater –a (default to be 10) to the same step size.** All files in these formats in the specified directory will be automatically taken as input data.

For paired-end input reads, we strongly suggest .bam or .sam formats. For **paired-end reads in .bed format**, the names of the two reads of each paire must be the same except for the last letter in each name, and the two reads must be located next to each other in the file, e.g.

…
chrIX 214478 214553 read45214/1 0 +
chrIX 214628 214703 read45214/2 0 -
…

OUTPUTS

All result files and will be in a directory named by the parameter **–o** and its subdirectories:

raw/ This directory contains .wig format nucleosome raw occupancy for each replicate. The occupancy values are eitherread from input .wig file or calculated from raw reads. This directory will be created only when the parameter –k is set to 1.

bgsub/ This directory contains .wig format nucleosome occupancy data from each replicate, from which the genomic background has been subtracted. This directory will be created only when the parameter –k is set to 1 and –b is provided.

nor/ This directory contains normalized .wig format nucleosome occupancy data for each replicate. This directory will be created only when the parameter –k is set to 1 and –n is not set to 'N'.

scaled/ This directory contains nucleosome occupancy data that has been scaled to the level specified by the parameter –c. This directory will be created only when the parameter –k is set to 1 and –c is provided.

ajClonal/ This directory contains nucleosome occupancy data in which clonal signal has been adjusted. This directory will be created only when the parameter –k is set to 1 and –clonalcut is not set to 0.

smoothed/ This directory contains smoothed .wig format nucleosome occupancy data for each replicate. This directory will be created only when the parameter –k is set to 1 and –z is not set to 0.

replicate_peaks/ This directory contains nucleosome occupancy peaks called for each replicate. This directory will be generated only when the parameter –k is set to 1 and –x is set to 1.

pooled/ This directory contains one nucleosome occupancy file for each group, replicates from the same group are pooled into a single occupancy file. Nucleosome occupancy peaks called for each group are also stored in this directory.

diff/ This directory contains .wig format differential signal generated for each pair of comparison specified. Differential peaks were also saved in this directory.

***.peaks.xls** occupancy peaks (nucleosomes) or differential peaks (dynamic nucleosomes). There are 8 columns in the occupancy peak files, whereas the differential peak file has only the first 5 columns:

Chr: chromosome

Start: peak start position

End: peak end position

Smt_pos: peak summit position

Smt_value: peak summit value

smt_pval: P value of the summit value

fuzziness_score: peak fuzziness score

fuzziness_pval: P value of the peak fuzziness score

***.allPeaks.xls**

A final result file combining the occupancy peaks and differential peaks. Scores for single-nucleosome-resolution occupancy change, fuzziness change, and position shift are also provided in this file. There are 14 columns in this file:

Chr: chromosome

control_smt_loca: peak summit position in the control

treat_smt_loca: peak summit poison in the treatment

diff_smt_loca: summit position of differential peak

treat2control_dis: the shift distance between *control_smt_loca* and and *treat_smt_loca*

control_smt_val: occupancy value in the control at control_smt_loca

treat_smt_val: occupancy value at in the treatment at treat_smt_loca

0-log10smt_diff_pval: smt_diff_pval is the P value of difference between control_smt_val and treat_smt_val, 0-log10smt_diff_pval is 0-$\log_{10}$(smt_diff_pval)

smt_diff_FDR: the FDR of difference between control_smt_val and treat_smt_val

control_point_val: occupancy value in the control at diff_smt_loca

treat_point_val: occupancy value in the treatment at diff_smt_loca

0-log10point_diff_pval: point_diff_pval is the P value of difference between control_smt_val and treat_smt_val, 0-log10point_diff_pval is 0-$\log_{10}$(point_diff_pval)

point_diff_FDR: the FDR of difference between control_point_val and treat_point_val

control_fuzziness_score: nuclesome fuzziness score of the control peak

treat_fuzziness_score: nuclesome fuzziness score of the treatment peak

0-log10fuzziness_diff_pval: fuzziness_diff_pval is the P value of difference between control_fuzziness_score and treat_fuzziness_score, 0-log10fuzziness _diff_pval is 0-$\log_{10}$(fuzziness_diff_pval)

fuzziness_diff_FDR: the FDR of difference between control_fuzziness_score and treat_fuzziness_score

Frequently Asked Questions

1. How to classify nucleosomes bearing occupancy change, fuzziness change, and position shift?
   General, when we just want to analyze total dynamic nucleosomes, simply rank dynamic nuclesomes in the result file ***.allPeaks.xls** by

"**point_diff_FDR**", e.g., select the ones with "point_diff_FDR" lower than 1e-2. If need to classify dynamic nucleosomes into the three categories, the following additional steps are suggested:

(1) To retrieve dynamic nucleosomes bearing position shift, rank nucleosomes by "**treat2control_dis**" and select the dynamic nucleosome by an upper cutoff and lower cutoff, e.g., select nucleosomes with shift distance between 50 and 90 bp and with "point_diff_FDR" lower than 1e-2.

(2) To retrive dynamic nucleosomes bearing fuzziness change, rank the dynamic nucleosomes by "**fuzziness_diff_FDR**" , e.g., select the ones with "fuzziness_diff_FDR" lower than 1e-2 and with "point_diff_pval" lower than 1e-2.

(3) To retrive occupancy changes, rank the dynamic nucleosomes by "**smt_diff_FDR**", e.g., select the ones with "smt_diff_FDR" lower than 1e-2 and "point_diff_FDR" lower than 1e-2. Some time we may want to remove the subset that shows both low "smt_diff_FDR" and low "fuzziness_diff_FDR".

2. Why I have an extremely large list of differential nucleosomes with significantly low P values (e.g. very large 0-log10point_diff_pval) in the file ***.allPeaks.xls**?

   A lower P value means more significant difference. When we have a higher sequencing coverage (more reads), we will have a more significant P value, the rationale is that more sequencing will give us more confident on the result, e.g. a difference between 10 and 50 will have lower P value than that between 1 and 5.

   Therefore, at a given P value cutoff, higher sequencing coverage will let you have larger count of differential nucleosomes. We suggest a P value cutoff equal to or lower than 1e-5, so you are free to select a more stringent cutoff, e.g. 1e-50, or some time even 1e-300, as long as the selected differential nucleosomes make sense to you. In some case you may also combine the P value with Fold change between control_point_val and treat_point_val to further select your candidate differential nucleosomes. The FDR also provide an option to further reduce false positive observation.

3. How about the speed and memory cost of DANPOS?

   The memory cost depends on the genome size and sample replicates count,

   Memory ~ (genome_size/step_size) x (replicate_count + max(2, comparison_count)) x 8 bits.

   For example, for a genome with size at 1G base pairs, if we have 1 sample and 1 replicate, so there's 0 comparisons, step size will be 10 bp by default, so the memory cost is close to:

   $(1 \times 10^9/10) \times (1 + \max(2, 0)) \times 8$ bits = 2.4 G bits

if we have 2 groups A, B, each group has 1 replicate, and 1 comparisons A:B, step size will be 10 bp by default, so the memory cost is close to:
$(1 \times 10^9/10) \times (2 + \max(2, 1)) \times 8$ bits = 3.2 G bits
if we have 3 groups A, B, and C, each group has 1 replicate, and we are doing 2 comparisons A:B and A:C, step size will be 10 bp by default, so the memory cost is close to:
$(1 \times 10^9/10) \times (3 + \max(2, 2)) \times 8$ bits = 4 G bits

The time cost depends on the reads count and genome size.

Time ~ reads_count x disk_speed + (group_count + comparison_count) x CPU_speed

For yeast genome, job can always be done in several minutes. For human genome, each comparison may take several hours, thus we suggest doing each comparison individually in a parallel way for large genomes when there's enough memory. For example:

Python danpos.py A:B,A:C

can be replaced by:

Python danpos.py A:B
Python danpos.py A:C

4. Is there any way to run DANPOS when my system doesn't have enough memory for large genomes such as the human genome?
   Two ways will be helpful while not change the final results:
   (1) Transform reads data of each replicate to occupancy data first:
   python danpos.py reads_data.bed
   And then use all replicates of occupancy data as input to do occupancy normalization, differential test, and peak calling:
   (2) Split the data by chromosomes and run a subset of chromosomes each time.

5. May I use DANPOS to analyze ChIP-seq data such as histone modifications or histone variants?
   Most previous analysis of ChIP-seq data was interested in enriched regions with variable sizes from dozens to thousands of bas pairs, but always didn't carry out analysis at each individual nucleosome. For such task, we suggest to use DyChIPS instead of DANPOS. DyChIPS is provided at the project website http://code.google.com/p/dychips/

   We have not tested DANPOS on ChIP-seq data, but we do agree that analyzing ChIP-seq data for each invidual nucleosome would be a interesting topic, it will be one of our major efforts in the future improvement of DANPOS. There, we also encourage users to test DANPOS on ChIP-seq data; we will appreciate if you are willing to provide suggestions on this point.

   Since most ChIP-seq data has an average fragment size larger than 200bp, so for each fragment, if one end comes from one edge of a nucleosome, then the other end will probably not be from the other edge of the same nucleosome.

This makes it hard to distinguish between nucleosome dyad and linker, and thus hard to analyze each individual nucleosome. Therefore, if our users would like to use DANPOS to analyze ChiP-seq data for each nucleosome, we strongly suggest making sure that the DNA fragments in the ChIP-seq experiment are largely from mono-nucleosomes, this means the distribution of fragment sizes should be around 147bp.

6. Is there a way to run the pipeline faster when I have multiple pairs of comparisons?
When we have 3 data sets a, b, and c, and want to do comparisons a:b, b:c, and a:c. We may hope that the comparisons themselves are comparable to each other, e.g. is the count of differential nucleosomes in a:b more than that in a:c? So it is important to normalize a, b, and c to the same level before the comparisons.

   By the following command:

   python danpos.py a:b,b:c,a:c

   DANPOS will normalize a, b, and c to the same level and do the comparisons one by one.

   For small genomes such as the yeast genome, the cost of time will not be an issue for DANPOS. However, when we have more data sets for large genome, it may take longer than a day to finish the analysis. As an instead, we can use DANPOS in three separate steps:
   (1) Calculate raw occupancy from mapped reads by the command:

   python danpos.py a
   python danpos.py b
   python danpos.py c

   (2) Move the occupancy data to new directories oa, ob, and oc, then do normalization by the following command:

   python danpos.py oa,ob,oc

   (3) Finially, move the normalized occupancy data to new directories na,nb, and nc, then do comparisons by the following commands:

   python danpos.py na:nb
   python danpos.py nb:nc
   python danpos.py na:nc

REFERENCES

1. N. Kaplan, I. K. Moore, Y. Fondufe-Mittendorf et al., *Nature* **458** (7236), 362 (2009).
2. Xing-sheng Shu, Hua Geng, Lili Li et al., in *PLoS ONE* (Public Library of Science, 2011), Vol. 6, pp. e27346.
3. Y. Zhang, H. Shin, J. S. Song et al., *BMC Genomics* **9**, 537 (2008).
4. Y. Xi, J. Yao, R. Chen et al., *Genome Res* **21** (5), 718 (2011).
5. S. J. Robinson and I. A. Parkin, *BMC Genomics* **9**, 434 (2008).

6. M. D. Robinson, D. J. McCarthy, and G. K. Smyth, *Bioinformatics* **26** (1), 139 (2009).