A pipeline for ChIP-seq data analysis (Prot 56)

Ruhi Ali[1], Florence M.G. Cavalli[1], Juan M. Vaquerizas[1] and Nicholas M. Luscombe[2,3,4].

1. European Bioinformatics Institute. Wellcome Trust Genome Campus, Cambridge CB10 1SD, UK. 2. Okinawa Institute of Science & Technology, 1919-1 Tancha, Onna-son, Kunigami- gun, Okinawa 904-0495, Japan. 3. University College London Genetics Institute, Gower Street, London WC1E 6BT, UK. 4. CRUK London Research Institute, 44 Lincoln's Inn Fields, London WC2A 3LY, UK

Publication Date: 24 May 2012

Last reviewed: 3 April 2012 by Gareth A Wilson
Medical Genomics Group, UCL Cancer Institute, University College London

Introduction

Chromatin immunoprecipitation followed by sequencing (ChIP-seq) is becoming the standard experimental procedure to investigate transcriptional regulation and epigenetic mechanisms on a genome-wide scale (reviewed in (Park, 2009)). The technique involves covalent cross-linking of proteins to the DNA, followed by fragmentation and immunoprecipitation (IP) of the chromatin by using an antibody against the protein or histone modification of interest. The result of this experiment is a set of short DNA fragments of about 200 bp in length that represent regions of the genome where the protein is bound, or where specific histone modifications occurred. The segments are then sequenced using one of the various next generation sequencing procedures now available. The resulting reads (usually 36 to 100bp) are then mapped back to the reference genome of interest in order to identify regions with significant binding.

Since the introduction of the experimental technique, several bioinformatics approaches have been developed to cope with the analysis of these data (reviewed in (Laajala et al., 2009; Wilbanks and Facciotti, 2010)). Usually these different methods have been initially developed to analyse a given dataset and associated experimental design and therefore they are based on different assumptions. For example, some methods used window-based scans to establish read density profiles, while others use different kernel density estimators; some perform peak assignments in a strand-specific basis, whereas others do so in a non-strand sensitive fashion; some implement the usage of control or background datasets; and, usually, every method is based on a different statistical model or test and uses alternative approaches to adjust for multiple testing or to normalise the data. Given such disparity in methods, it is not difficult to imagine that the results obtained from such analyses are heavily dependent on the method employed, with peak overlaps ranging from 100 to 13% depending on the algorithm (Wilbanks and Facciotti, 2010).

Here, we present a step-by-step protocol for the analysis of ChIP-seq data using a new robust procedure based on the estimation of background signal using an input DNA control. Unlike many of the currently available methods, which are based on fitting the ChIP-seq signal to a given distribution, our approach is based on an unbiased evaluation of the noise in the sample that is then used to calculate the statistical significance of the binding events. Hence, our procedure is ideal for profiles where no previous information about the mode of binding –e.g. sharp peaks or broad domains– is known. The method, implemented through the statistical package R/Bioconductor (Gentleman et al., 2004), has been successfully used for small genomes such as *D. melanogaster* (Schwartz et al., 2006; Kind et al., 2008; Conrad et al., 2012), and can be used for any dataset with a sufficient coverage for both the input and the IP sample. In this protocol, we use a recent ChIP-seq dataset by Raja et al. to illustrate each step of the analysis (Raja et al., 2010). The code is available at http://www.ebi.ac.uk/luscombe-srv/protocols/epigenesys.

Overview of the pipeline

Briefly, the protocol uses the distribution of log fold-changes between the IP and an input sample to estimate the background noise of the experiment. This is done by calculating a symmetric null distribution using fold-change values where the signal obtained through the input sample is higher than that from the IP sample (Schwartz et al., 2006). Given that in this type of experiment a biological enrichment should only appear for genomic regions where the assayed TF-binding or chromatin modifications occur, the enrichments found in the input

sample serve as a good indicator of systematic non-biological biases that can be used to estimate the background noise. After calculating the distribution of systematic experimental noise, a p-value is calculated for each genomic region, resulting in a list of regions where the enrichment of the IP sample is significant.

Counting mapped reads

As input, the protocol requires a set of IP and input reads mapped to the corresponding reference genome. Prior to mapping, we encourage users to perform a quality control and any needed pre-processing of their sequencing data using any of the publicly available procedures (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/; Patel and Jain, 2012; Planet et al., 2012). Then, reads can be mapped using any of the alignment software publicly available, such a Bowtie, MAQ or BWA (Li et al., 2008; Langmead et al., 2009; Li and Durbin, 2009). In this protocol we use Bowtie (version 0.12.17). Bowtie implements many parameters regarding specific protocols to map reads to a reference genome, and explaining its usage is out of the scope of this protocol. However, we invite you to read Bowtie's documentation (http://bowtie-bio.sourceforge.net/index.shtml) for further details and for installation instructions.

In order to map reads back to the reference genome, and given a file called 'reads.fastq' containing your reads in FASTQ format, the following command line will map reads to the reference genome returning those that map uniquely:

```
ebi-001:~ ruhi$ ./bowtie -a -m 1 -q d_melanogaster_fb5_22 reads.fastq -S
```

The –a option specifies to report all alignments and the option –m to remove all alignments that map to more than one location. The argument –q specifies that the file is in a FASTQ format, 'd_melanogaster_fb5_22' is the base-name of the index, which contains the information about the reference genome to be used, and –S specifies the alignments to be returned in SAM format. Indices for many model organisms are available at the Bowtie website or can be readily built by the user. For more information about various other parameters implemented by Bowtie, please visit: http://bowtie-bio.sourceforge.net/index.shtml.

Once reads have been mapped back to the reference genome, they need to be converted into BAM format, if the alignment software has not already done this. For example, when

using Bowtie the alignments will be returned in SAM format. These can be easily transformed into BAM using the following SAMtools command:

```
ebi-001:~ ruhi$ samtools view -bS -o output.bam input.sam
```

where input.sam is the SAM file containing all the alignments created by bowtie. In order to be able to run this command, you should have the SAMtools suite of tools installed on your system (Li et al., 2009).

Extending mapped reads

Usually the reads returned from the sequencer will be between 36-100 bp. The first step of the analysis consists of a read extension to the average size of fragment length in the original chromatin preparation. The objective of the extension is to maximise the coverage for each fragment. Otherwise, given that the DNA fragments are sequenced only from the 5'-end of each strand and that this is done independently for each strand, not extending the reads to the expected fragment length might result in artefacts such as double peaks in between small troughs in some areas (Park, 2009). To do this, first we divide the genome of the organism in equally sized bins of 25 bp (this number is arbitrary and the user can select a different binning strategy depending the requirements of the analysis). Then we calculate the number of reads that overlap a bin, taking into account that the length of the read corresponds to the average fragment size. This is done independently for each sample, both immunoprecipitated (IP) and input (or control) ones, and for each biological replicate, when available. Future versions of the pipeline will include support for paired-end reads, where the read extension is not required, as the size of the fragment is known.

Normalizing the read counts and calculating a log2 fold-ratio

Next, in order to account for differences in sequencing depth, the samples are normalised using DESeq (Anders and Huber, 2010). Briefly, DESeq calculates the geometric mean of reads in the experiment and then a factor for each sample that corrects for variations in sequencing depth. More details about DESeq can be found here: http://www-huber.embl.de/users/anders/DESeq/.

Once all samples have been normalised for sequencing depth, we calculate a log2 fold-ratio between the IP and the input or control sample. Using input as a control sample is essential

for this analysis: (i) it will provide us with a baseline of noise, that we will use to estimate the background and therefore the enrichment of the IP sample; and (ii) it helps in correcting systematic biases such as differential shearing of open and closed chromatin regions – since these should be found both in the input as well as in the IP sample; using this matched control sample will allow us to remove these biases (Park, 2009). If the experimental design involves duplicates, the current version of the pipeline will treat them as independent samples and will calculate an average value per bin without using information about the variability between replicates. Future versions of the pipeline will implement different options to include replicate variability information.

Smoothing over a specified window size

Once the log2 fold ratio has been calculated we apply a smoothening approach to the data. To do so, for each bin in the genome, we calculate a smoothed value using the log2 fold ratios overlapping a surrounding window of 2**n* bp centred in the original position, where n is the average size fragment in the experiment (eg, 200bp). The smoothed value can be calculated by multiplying the rolling average of the log2 fold ratios in a given window by the square root of the number of bins that have a numerical value (to correct for the number of bins in regions that do not have any mapped reads, for example due to the presence of repetitive regions). The rationale for this approach is that if the fragment size is longer than the bins in which we have divided the genome, true binding will necessarily overlap more than one bin. Therefore, by smoothing we contribute to the robustness of the data by integrating the signal for the entire locus.

At this stage we can perform a visual analysis of the process by plotting the distribution of log2 fold-change. A density plot of the smoothed values should look similar to a normal distribution centred around zero with a pronounced right-hand side tail (Figure 1). The 'shoulder' in the right-hand side of the distribution corresponds to the enrichment of the IP sample. Distributions that deviate from this pattern will most likely indicate a problem with the sample. For example, lack of enrichment in the right-hand side might indicate poor IP enrichment. Similarly, plots displaying non-smooth distributions will indicate regions of the genome with low coverage or an overall low overlap between IP and input sample. The visual inspection of these plots is crucial for the assessment of the results.

Computing loci with significant binding events

As observed in Figure 1, the left-hand side of the distribution, corresponding to those bins where the input sample is higher than the IP, should have no obvious enrichment. Given that the input sample should not contain any specific enrichment other that those caused by differential shearing, etc., we use these values to estimate the systematic variability found in the sample. To do so, we calculate a symmetric-null distribution based on values below the mode of the log2 fold-change (Figure 2). This distribution is then used to calculate p-values for the significance of enrichment compared with the symmetric-null distribution. p-values are then adjusted for multiple testing using FDR and bins with a FDR-adjusted p-value < 0.05 are selected as significant. Finally, consecutive windows with significant values are merged into binding regions, which are then reported in BED format.

Procedure

Installation of R and BioC packages

The method is currently implemented in the statistical programming framework R (version 12.0 or above). For instructions on how to install R, please visit the following location: http://cran.r-project.org/mirrors.html.

The pipeline requires the following three packages from BioConductor: ShortRead, DESeq and GenomicRanges. These can be installed using following commands from the R command prompt.

```
source("http://bioconductor.org/biocLite.R")
biocLite("ShortRead")
biocLite("DESeq")
biocLite("GenomicRanges")
```

An extra package is required, symp, which can be downloaded from the protocol website (http://www.ebi.ac.uk/luscombe-srv/protocols/epigenesys/) and installed by running the following command in a terminal:

```
ebi-001:~ ruhi$ R CMD INSTALL symp.tar.gz
```

After the installation finishes the packages can be loaded using the following commands:

```
library(ShortRead)
library(DESeq)
library(GenomicRanges)
library(symp)
```

Example dataset: NSL1 ChIP-seq data

In order to demonstrate the performance of the method, we will use publicly available ChIP-seq data for NSL1 (Raja SJ et al). NSL1 is part of the NSL complex, which acts as a major transcription regulator in *Drosophila.* The data for the NSL1 immunoprecipitation and input samples can be downloaded from the following location: http://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-214.

These files contain 36 bp reads which have to first be mapped to the reference genome (see above), and then supplied in BAM format to this pipeline.

Running the code

After installing the packages, the pipeline will run the analysis using just one function `getSignificantBinding`:

```
getSignificantBinding("sample-annotation.txt","NSL1/input",
  Chromosome_list=c("chr2L","chr2R","chr3L","chr3R","chr4","chrX"),
  Read_size=36, Bin_size=25, Window_size=200, Significance_level=.05)
```

The function requires the mapped reads in BAM format and a set of variables that will be passed as arguments. In addition, the package needs two extra files indicating: (i) the set of reads to use in the analysis; and (ii) the information about the chromosome length for the required species.

Sample information. The annotation for the different files to use in the analysis should be stored as a tab-separated text file. The file should include two columns: the first one, named 'Filename', contains the path of the BAM file corresponding to a sample; the second column,

named 'Sample', must indicate an identifier for the sample, with all biological replicates annotated with the same identifier. For example:

```
ebi-001:~ ruhi$ more sample-annotation.txt
Filename  Sample
NSL1.bam  NSL1
Input.bam Input
```

Chromosome information. Chromosome information for the particular species has to be stored in a file called "*Org_Ch_length*", saved in the working directory. The chromosome information consists of the chromosome names as well as the length of each chromosome for the particular organism under analysis. In order to obtain the above information in the required format, download the particular BSgenome package from BioConductor. For *Drosophila*, this can be done by starting R and issuing the following commands:

```
source("http://bioconductor.org/biocLite.R")
biocLite("BSgenome.Dmelanogaster.UCSC.dm3")
```

Once the package is installed, chromosome lengths can be obtained and saved in a file with the commands:

```
library(BSgenome.Dmelanogaster.UCSC.dm3)
Org_Ch_length = seqlengths(Dmelanogaster)
save(Org_Ch_length, file="Org_Ch_length")
```

Here, it is important to make sure that the chromosomes have been named using the same chromosome identifier as when mapping the reads – for our example analysis this can be easily done by adding the prefix "chr" to the mapped reads before transforming them into BAM format:

```
ebi-001:~ ruhi$ sed 's/\t2L\t/\tchr2L\t/g' input.sam > new_input_2L.sam

ebi-001:~ ruhi$ sed 's/\t2R\t/\tchr2R\t/g' new_input_2L.sam >
  new_input_2L_2R.sam

ebi-001:~ ruhi$ sed 's/\t3L\t/\tchr3L\t/g' new_input_2L_2R.sam >
  new_input_2L_2R_3L.sam

ebi-001:~ ruhi$ sed 's/\t3R\t/\tchr3R\t/g' new_input_2L_2R_3L.sam >
  new_input_2L_2R_3L_3R.sam

ebi-001:~ ruhi$ sed 's/\t4\t/\tchr4\t/g' new_input_2L_2R_3L_3R.sam >
  new_input_2L_2R_3L_3R_4.sam
```

```
ebi-001:~ ruhi$ sed 's/\tX\t/\tchrX\t/g' new_input_2L_2R_3L_3R_4.sam >
  new_input_2L_2R_3L_3R_4_X.sam
```

Once the required files have been prepared, the user is ready to run `getSignificantBinding`. The function takes the following extra arguments:

i) Table_with_filenames: Location for the file containing the sample annotation.

ii) Comparison_vector: A vector indicating the two samples to be compared separated by "/". For example "NSL1/input" will calculate log2 fold changes of the IP versus the input sample.

iii) Chromosome_list: A vector containing the list of chromosomes exactly as they are named in the chromosomal information created above and in the mapped reads. For our analysis of the NSL1 data, these would be:

```
Chromosome_list = c("chr2L", "chr2R", "chr3L", "chr3R", "chr4", "chrX")
```

iv) Read_size: This is the size of the mapped reads.

v) Bin_size: The size of bins used to divide the genome (defaults to 25).

vi) Window_size: This argument is used to indicate the length of the DNA fragment and corresponds to half of the size that will be used to smooth the data. For example, a value of 200 will result in values smoothed over a window of 400 bp centred in the middle bin. The default value is 200 bp.

vi) Significance_level: The level of FDR-adjusted significance at which a bin will be called significant. Contiguous bins will be merged into regions of significant binding.

Results

Running the function with the appropriate arguments will result the in following results:

```
getSignificantBinding("sample-annotation.txt","NSL1/input",
  Chromosome_list=c("chr2L","chr2R","chr3L","chr3R","chr4","chrX"),
  Read_size=36, Bin_size=25, Window_size=200, Significance_level=.05)
```

i) A density plot of the smooth log2 fold change values (Figure 1) and a similar plot displaying the null distribution and the threshold used to determine significant binding (Figure 2).

ii) Two .sgr files corresponding to the smooth log2 fold change values across the genome for all bins and for bind with significant binding only. These files can be directly imported and visualised in genome browsers such as IGB.

iii) A .bed file containing all regions with significant binding. This is the result of a merge of consecutive bins with significant binding. This file can be used for further analysis or for visualising purposes.

iv) A .R file with chromosome names, genomic position, log2 fold change, smooth log2 fold change values for the different bins. In addition, the table contains an extra column where the log2 fold change is substituted by zero if the binding at that bin is not significant.

Figures

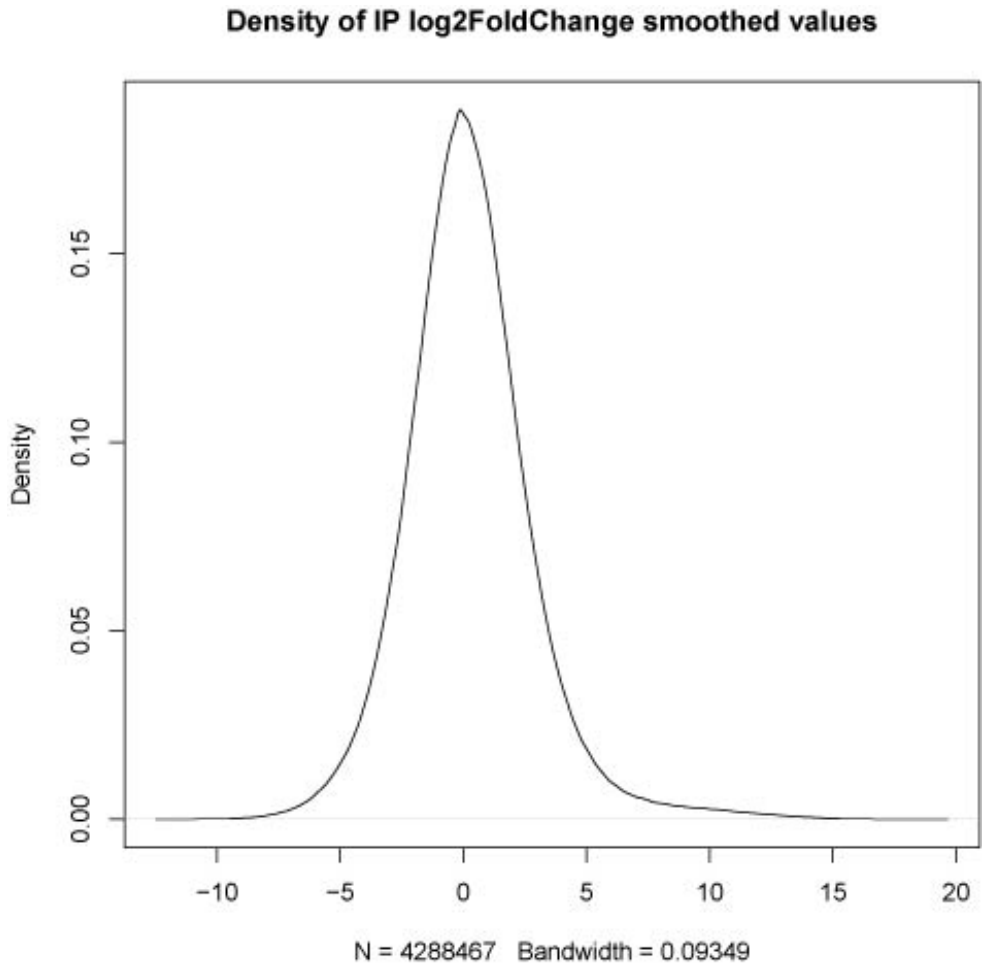

Figure 1. Density plot of the smooth log2 fold change values.

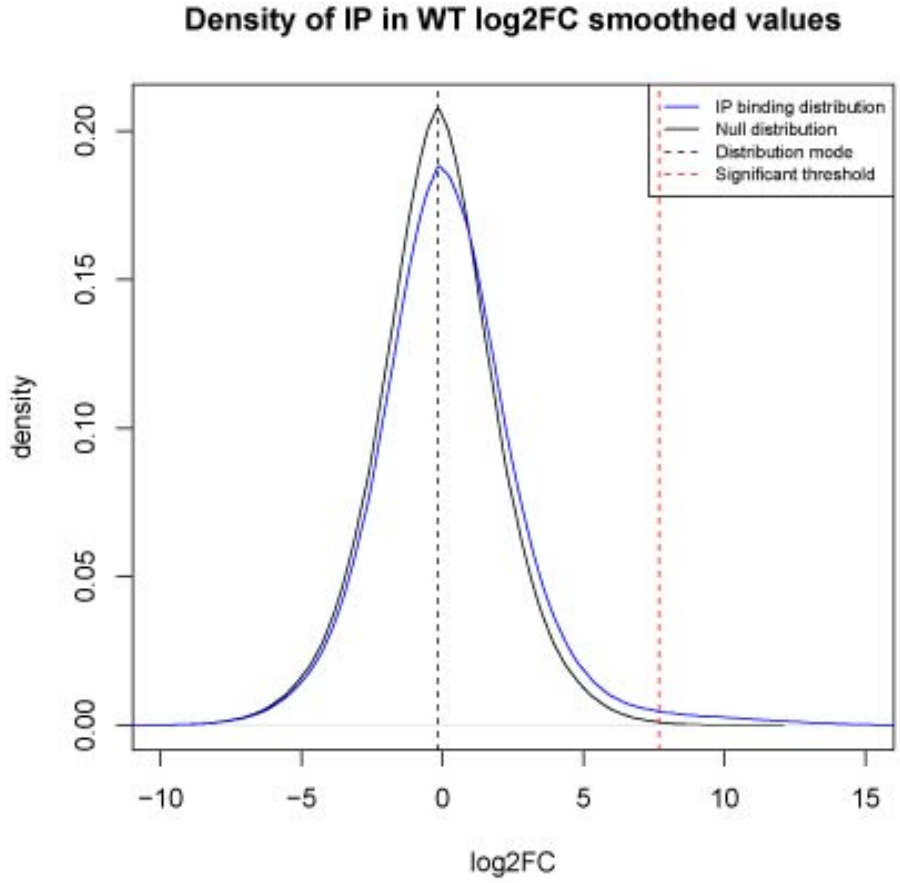

Figure 2. Density plot of the smooth log2 fold-change values (blue) plotted together with the symmetric null distribution (black). The mode used to compute the symmetric null distribution is indicated with a dotted black line. The threshold for significance is highlighted in red.

Reviewer comments:

Review by: Gareth A Wilson
Medical Genomics Group
UCL Cancer Institute
University College London

The protocol includes several modifications suggested by the reviewer.

References

Anders, S., and Huber, W. (2010). Differential expression analysis for sequence count data. Genome Biol. *11*, R106.

Conrad, T., Cavalli, F.M.G., Holz, H., Hallacli, E., Kind, J., Ilik, I., Vaquerizas, J.M., Luscombe, N.M., and Akhtar, A. (2012). The MOF Chromobarrel Domain Controls Genome-wide H4K16 Acetylation and Spreading of the MSL Complex. Dev. Cell *22*, 610–624.

Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., et al. (2004). Bioconductor: open software development for computational biology and bioinformatics. Genome Biol *5*, R80.

http://www.bioinformatics.babraham.ac.uk/projects/fastqc/ Babraham Bioinformatics - FastQC.

Kind, J., Vaquerizas, J.M., Gebhardt, P., Gentzel, M., Luscombe, N.M., Bertone, P., and Akhtar, A. (2008). Genome-wide analysis reveals MOF as a key regulator of dosage compensation and gene expression in Drosophila. Cell *133*, 813–828.

Laajala, T.D., Raghav, S., Tuomela, S., Lahesmaa, R., Aittokallio, T., and Elo, L.L. (2009). A practical comparison of methods for detecting transcription factor binding sites in ChIP-seq experiments. BMC Genomics *10*, 618.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S.L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol. *10*, R25.

Li, H., and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics *25*, 1754–1760.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., and Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. Bioinformatics *25*, 2078–2079.

Li, H., Ruan, J., and Durbin, R. (2008). Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res. *18*, 1851–1858.

Park, P.J. (2009). ChIP-seq: advantages and challenges of a maturing technology. Nat. Rev. Genet. *10*, 669–680.

Patel, R.K., and Jain, M. (2012). NGS QC Toolkit: a toolkit for quality control of next generation sequencing data. PLoS ONE *7*, e30619.

Planet, E., Attolini, C.S.-O., Reina, O., Flores, O., and Rossell, D. (2012). htSeqTools: high-throughput sequencing quality control, processing and visualization in R. Bioinformatics *28*, 589–590.

Raja, S.J., Charapitsa, I., Conrad, T., Vaquerizas, J.M., Gebhardt, P., Holz, H., Kadlec, J., Fraterman, S., Luscombe, N.M., and Akhtar, A. (2010). The nonspecific lethal complex is a transcriptional regulator in Drosophila. Mol. Cell *38*, 827–841.

Schwartz, Y.B., Kahn, T.G., Nix, D.A., Li, X.-Y., Bourgon, R., Biggin, M., and Pirrotta, V. (2006). Genome-wide analysis of Polycomb targets in Drosophila melanogaster. Nat. Genet. *38*, 700–705.

Wilbanks, E.G., and Facciotti, M.T. (2010). Evaluation of algorithm performance in ChIP-seq peak detection. PLoS ONE *5*, e11471.