# A computational pipeline for comparative ChIP-seq analyses

Anaïs F Bardet[1], Qiye He[2], Julia Zeitlinger[2,3] & Alexander Stark[1]

[1]Research Institute of Molecular Pathology, Vienna, Austria. [2]Stowers Institute for Medical Research, Kansas City, Missouri, USA. [3]Department of Pathology, University of Kansas Medical School, Kansas City, Kansas, USA. Correspondence should be addressed to J.Z. (jbz@stowers.org) or A.S. (stark@starklab.org).

Published online 15 December 2011; doi:10.1038/nprot.2011.420

**Chromatin immunoprecipitation (ChIP) followed by deep sequencing can now easily be performed across different conditions, time points and even species. However, analyzing such data is not trivial and standard methods are as yet unavailable. Here we present a protocol to systematically compare ChIP-sequencing (ChIP-seq) data across conditions. We first describe technical guidelines for data preprocessing, read mapping, read-density visualization and peak calling. We then describe methods and provide code with specific examples to compare different data sets across species and across conditions, including a threshold-free approach to measure global similarity, a strategy to assess the binary conservation of binding events and measurements for quantitative changes of binding. We discuss how differences in binding can be related to gene functions, gene expression and sequence changes. Once established, this protocol should take about 2 d to complete and be generally applicable to many data sets.**



## INTRODUCTION

To understand how *cis*-regulatory elements determine gene expression, the global identification of *in vivo* transcription factor binding sites is an invaluable tool. It is usually achieved by ChIP followed by microarray analysis (i.e., ChIP-chip)[1,2], or, more recently, by deep sequencing (ChIP-seq)[3,4]. The focus of many current ChIP-seq studies is the comparison of transcription factor binding profiles across different conditions such as different developmental time points[5,6], cell types (e.g., within one cell lineage[7,8]) or closely related species[9,10]. However, such comparative ChIP-seq studies are highly dependent on appropriate computational approaches, which are often still lacking. Most notably, stringent thresholds are typically used to reliably identify transcription factor binding sites. However, this method does not discriminate subthreshold binding from truly nonbound regions, and it is subject to noise, which can lead to an underestimation of the overlap in binding between two data sets.

Here we present a computational approach for the comparative analysis of ChIP-seq data that we recently developed to compare binding of the mesodermal transcription factor Twist across six closely related *Drosophila* species[9] (**Fig. 1**). We describe technical guidelines and provide code with sample data for the preprocessing and mapping of ChIP-seq reads, the translation of ChIP-seq data to a common reference genome (for cross-species analyses), approaches for a threshold-free comparison of global binding similarity, an analysis of binary presence/absence binding of patterns (e.g., to estimate the conservation of binding) and the assessment of quantitative changes in binding. We also discuss functional and comparative sequence analyses of transcription factor binding. Although this protocol was specifically developed for analyzing transcription factor ChIP-seq experiments in different *Drosophila* species[9], we have found that it works well when comparing transcription factor ChIP-seq data between different vertebrates and across different conditions (see ANTICIPATED RESULTS). We believe that the protocol can easily be adapted to ChIP-chip data or comparative studies of chromatin marks.

### Translation to common coordinates for cross-species comparisons

Comparative ChIP-seq analyses across different species require the data to be translated across genomes. Although a gene-centric approach is conceivable, it would restrict the analysis to genomic regions in the vicinity of genes. Therefore, when closely related species are analyzed, the easiest way is to translate species-specific genomic coordinates to a common reference (using available genome alignments and tools for coordinate translation such as LiftOver from the University of California Santa Cruz (UCSC)). The common reference species is typically the one with the most complete genome assembly and annotation, which is *Drosophila melanogaster* when comparing *Drosophila* species[9,11] and humans when comparing mammals or vertebrates[10,12,13]. We generally find that using a common reference genome works well in comparative ChIP-seq analyses, and that the measured binding divergence is mostly independent of the chosen reference genome as long as a similar number of peaks are identified in each sample[9] (see ANTICIPATED RESULTS).

There are two ways to translate ChIP-seq data to a common reference genome using the UCSC LiftOver tool. First, peaks can be called in the different species independently and the peak region coordinates then translated to the reference genome. Second, the raw reads can be mapped to the different respective genomes and their coordinates then directly translated to reference coordinates. Thus, the read coordinates rather than the peak coordinates are translated. We use the latter approach as we did not find substantial differences between the two approaches[9], and this approach allows a larger variety of downstream analyses (e.g., the assessment of global similarity by the Pearson correlation coefficient (PCC) and the analysis of quantitative changes).

### Assessing the global similarity of transcription factor binding

A powerful method to assess the overall similarity between two transcription factor binding landscapes is the PCC between the respective genome-wide read densities (read counts at each

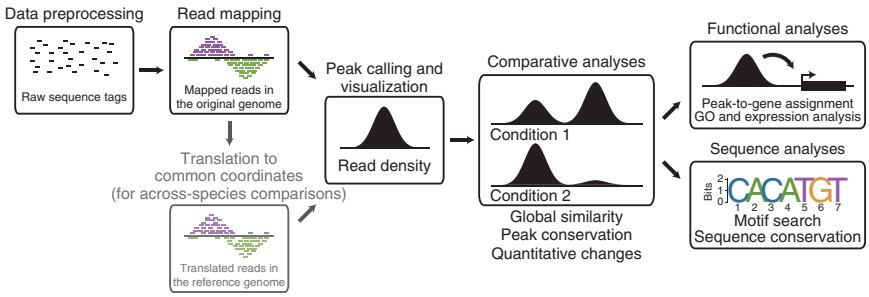


**Figure 1** | Computational pipeline for comparative analyses of ChIP-seq data. Raw reads are preprocessed and mapped to the respective genome sequences. For comparisons across species, mapped reads are translated onto a chosen reference genome. Read densities can be visualized along the genome, and peaks representing binding events are called. Comparative analyses include a threshold-free comparison of global binding similarity, analyses of the binary presence/absence of binding patterns (i.e., peak conservation) and quantitative assessment of binding changes. Functional and sequence analyses such as expression and Gene Ontology[33] (GO) analysis of target genes, motif search and sequence conservation can then be conducted.

position in the genome). As the PCC is threshold independent and invariant to scale, it eliminates some of the challenges associated with using thresholds for peak calling, and is more robust against experimental variation of peak heights. We also use the PCC to assess the similarity between biological replicates and to obtain a quality measure of each ChIP sample by comparing it with the corresponding input control sample (see Experimental design and Anticipated results).

### Global identification of transcription factor-binding sites ('peak calling')

A common step in all ChIP-seq analyses is the global identification of transcription factor binding sites or 'peaks', which are regions with markedly enriched read densities in the ChIP sample. Many computational tools are available for calling peaks reliably in the entire genome (e.g., MACS[14]; for an overview see ref.15). Typically, enrichments of read counts are calculated between the ChIP sample and an input sample, which should control for potential biases in the experimental procedure (see Experimental design). Another important element is the correction for multiple testing, as the peaks are selected by testing a large number of possible genomic regions for high ChIP enrichment; i.e., a scenario in which even the best of many random candidates would show good enrichments[16,17]. A good measurement that corrects for multiple testing is the false discovery rate (FDR; we recommend ≤ 1% when calling peaks). Note that most programs for ChIP-seq data analysis assess the FDR empirically, e.g., by swapping ChIP and input samples (i.e., MACS).

### Comparing peak presence across conditions (binary analysis)

Although calling peaks in a ChIP-seq sample is well established, comparing two ChIP-seq samples with each other is not. Merely comparing two samples by overlapping the genomic coordinates of their respective individually called binding peaks has inherent statistical problems, and leads to an underestimation of binding similarity (**Fig. 2**).

First, the overall global binding similarity is underestimated because of the so-called 'winner's curse'[18]. Genome-wide experiments are intrinsically subject to noise, and thus replicate experiments systematically produce different values or ranks for peaks, even if the samples are of very high quality. Therefore, if two replicates are independently thresholded at an identical value, peaks that are above the threshold in one might be below the threshold in the other and vice versa. Although this is appropriate to stringently define a high-confidence set of peaks, it prohibits a fair estimation of the respective number of peaks that are shared between conditions versus peaks that are condition-specific. For example, if we compare two replicate experiments by overlapping their binding peaks, we typically find that only ~75% of peak regions overlap with a peak region in the other sample (see also **Fig. 3** and ANTICIPATED RESULTS), although their binding profiles look virtually identical and show PCCs of 0.9 and higher.

Second, intersecting independently called peak regions are overly stringent as each sample is corrected for multiple testing during peak calling. When assessing binding across different conditions, one is generally interested in the number of shared and unique binding sites, a scenario in which significance measures must not be corrected for multiple testing: the task is not to assess the significance of the very best shared peaks, but rather to fairly assess the number of both types of peaks. The use of multiple testing correction makes the threshold for binding in the second data set too stringent and leads to an underestimation of shared binding events.

To address these issues, we do not intersect peak regions, but instead separate the steps of binding site identification from the analysis of binding site changes. Although we call peaks with a stringent multiple testing–corrected FDR threshold for the reference sample, we assess binding in the other samples by a nonrandom enrichment of ChIP versus input (not corrected for multiple testing) at the positions corresponding to each binding site in the reference sample. Using this protocol, we typically found a near

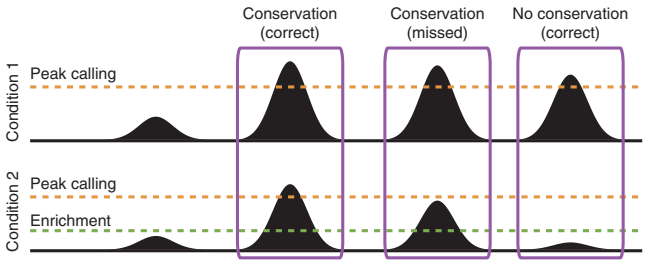


**Figure 2** | Choice of sensitive thresholds when comparing ChIP-seq samples. During genome-wide peak calling, only the best peaks pass the stringent thresholds required for low false discovery rates (FDRs) due to the correction for multiple testing (orange lines). Regions may show substantial tag enrichment, yet are not called as peaks (green line). When comparing peaks across conditions, we advocate using 'significant enrichment' (not multiple testing–corrected) as the measure to assess whether a peak is shared across conditions or is truly condition-specific. Merely intersecting peaks called at each condition would miss conserved peaks (e.g., middle examples).

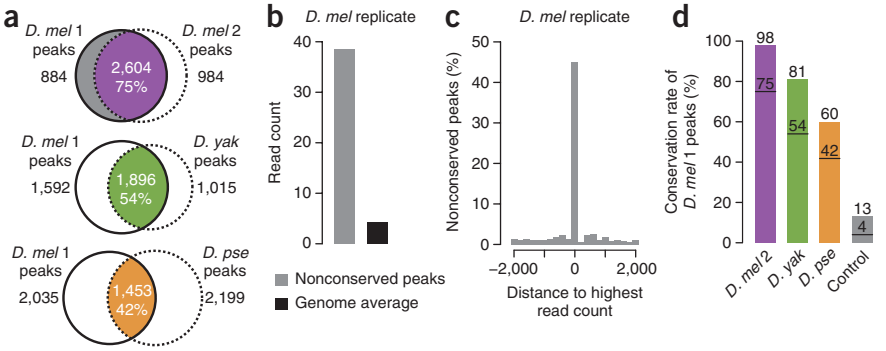


**Figure 3** | Assessing choice of thresholds and its impact on conservation estimates. (**a**) Conservation estimates based on overlapping high-confidence peaks: *D. melanogaster* versus *D. melanogaster* replicate (purple), *D. yakuba* (green) and *D. pseudoobscura* (orange). The nonconserved peaks between the *D. melanogaster* replicates (gray) highlight the problem inherent to this approach (**Fig. 2**). (**b**) The average read count of nonconserved replicate peaks (gray) from **a** is much higher than the genome average. (**c**) The highest read count within a 4-kb window around a peak of the reference data set that appears nonconserved in a biological replicate (gray, see **a**) remains at the position that corresponds to the peak summit of the reference. (**d**) By requiring high-confidence peaks to display a significant enrichment of read count in the other conditions, more sensitive conservation estimates (numbers above bars) are obtained for a biological replicate (purple), close species (*D. yakuba*; green) and more distant species (*D. pseudoobscura*; orange) compared with using an identical threshold in both species (black lines). Random control regions (gray) are obtained by offsetting all peaks by 20 kb. *D. mel*, *D. melanogaster*; *D. yak*, *D. yakuba*; *D. pse*, *D. pseudoobscura*. Data are from He *et al.*[9].



100% agreement between biological replicates while not substantially underestimating divergence as shown using control peaks (i.e., peaks shifted to random locations).

### Assessing quantitative changes in transcription factor binding

Transcription factor binding across a population of cells is not an all-or-none phenomenon, but rather represents a quantitative measure[19]; i.e., transcription factors can occupy their binding sites at different rates. To measure these more subtle quantitative binding differences across samples, the quantitative changes in peak heights can be analyzed across samples[11]. We first stringently call peaks for each of the conditions independently using multiple testing corrected thresholds and compile a unique set of peaks called in at least one condition. These positions are then used to assess the peak heights and the corresponding genomic positions under all conditions. Thus, peak heights are assessed even when the peak was not called under a specific condition. Note that as peaks from all conditions are analyzed, the identified changes in binding between conditions are inherently unbiased (i.e., symmetrical) with respect to the different samples and the choice of the reference sample.

For such analysis, a key consideration is the normalization method (see also EXPERIMENTAL DESIGN). When comparing conditions within one organism using the same antibody, we recommend normalizing only the read counts to the respective library sizes and input controls. This allows the comparisons of different conditions even when the total number and height of peaks is expected to change (e.g., an induced versus uninduced condition). When using different antibodies or different species, the signal-to-noise ratios might not be comparable across experiments because of differences in the antibodies' affinities. In this case, it is helpful if one can reasonably assume that the total number of binding sites is constant (e.g., when studying a conserved biological system across different species[9]). Furthermore, the heights of peaks and the corresponding genomic locations can be normalized using quantile normalization, a method that is frequently used in microarray data analysis.

### Functional analysis

A frequent goal of ChIP-seq experiments is the assignment of target genes to the binding peaks identified for transcription factors. This is nontrivial, however, as enhancers bound by transcription factors are able to activate their target genes from remote distances and even across nonregulated genes located in between (e.g., more than 1 Mb for the mouse gene *Shh* (encoding Sonic hedgehog)[20,21]; see also shadow enhancers in *Drosophila*[22]). Although such distances may not actually be far within the spatial arrangement of the chromatinized genome in the nucleus, information about 3D contacts between genomic regions is not available and cannot be used for peak-to-gene assignments. A practical shortcut is therefore to assign peaks to the closest gene transcription start site (TSS) along the genome sequence. As data on insulator protein binding sites are now available (e.g., for the *Drosophila*[23], mouse[13] and human[24] genomes), gene assignment can be prohibited across insulator sites.

Once peaks are assigned to target genes, the target genes can be functionally analyzed using Gene Ontology (GO) categories or gene expression data. This cannot easily be done by standard analyses as peak-to-gene assignment is heavily biased by gene lengths[25] (see discussion in ref. 26), which often leads similar categories to seem enriched in all samples. To solve this problem, we determine the rate of binding change between samples for all peaks in each GO category or expression class (i.e., the fraction of the conserved (or divergent) peaks among all peaks per GO category). In this manner, the analysis is independent of the overall number of peaks in each category.

### Comparative sequence analysis

Experimentally determined transcription factor binding across different species or different conditions provides an opportunity for analyzing the sequences that may mediate transcription factor binding. In fact, such comparative ChIP-seq data sets have proved successful in illuminating potential mechanisms of combinatorial binding[9–11,27,28]. This is because sequences in enhancers frequently change despite the conservation of enhancer function, but important binding motifs for transcription factors are often conserved (reviewed in ref. 29). Here we describe approaches for analyzing overall sequence conservation and divergence in binding regions (i.e., mutations, insertions and deletions), as well as for investigating the conservation of specific transcription factor binding motifs in peaks with binding loss, gain or quantitative change.

### Examples of data that can be analyzed with our procedure

The procedure has been developed for the comparative analysis of Twist ChIP-seq data from different *Drosophila* species[9], and we

provide an original raw data set so that our analysis steps can be traced and used as a guide (see MATERIALS). We have also tested the applicability of our comparative pipeline in vertebrate species, as well as in *Caenorhabditis elegans* binding data across different developmental stages. For vertebrates, we analyzed CEBPA binding in the livers of humans, mouse, dog, opossum and chicken[10], and found that our approach is sensitive across a wide range of thresholds. In *C. elegans*, we compared ChIP-seq data of the transcription factor PHA4/FOXA in embryo and the first stage of larval development[30].

### Experimental design

**General principle.** In a ChIP experiment, transcription factors are cross-linked to DNA in their native state and whole-cell extract is prepared, which serves as input for the immunoprecipitation[31]. During the immunoprecipitation, the transcription factor and the associated DNA fragments are pulled down from the extract. As some proteins and DNA fragments are also pulled down nonspecifically, the DNA fragments that are sequenced at the end are a mixture between real signal (the binding sites of the transcription factor) and nonspecific background. To achieve high signal-to-noise ratios, a good antibody is crucial. However, the amount of starting material and the exact experimental conditions can also influence the signal-to-noise ratios. After systematic optimization of the protocol, small variations may still exist between different experiments.

**Choice of a control sample.** To control for the nonspecific background, the input sample or sample obtained from a mock immunoprecipitation (the same procedure without specific antibodies) is sequenced. Although a mock immunoprecipitation is the ideal control in theory, it can produce DNA that is below the recommended amount for sequencing. Even if such low amounts of DNA can be amplified and sequenced, the sample may be noisy and unrepresentative as a result. For this reason, we use the input sample as control.

**Planning for data normalization.** As the signal-to-noise ratio in comparative ChIP-seq experiments may differ, we recommend following one of two strategies. First, if samples from different experimental conditions are compared and the same antibody is used[32], we recommend performing the series of experiments side by side as this minimizes differences in signal-to-noise ratios due to experimental variability. By using this strategy, the ChIP-seq data do not need to be normalized to each other (other than by the total library read count) and differences in overall binding enrichments can be detected. Second, if this strategy is not possible because different species or antibodies are used, quantile normalization can be used to adjust for differences in signal-to-noise ratios between samples if one can reasonably assume that the overall binding of the factor is similar (e.g., if the factor is well conserved and is expressed at similar levels in the same tissue across species). If this assumption is not justified, it is still possible to identify qualitative differences between samples while being aware that conclusions on the overall binding strength cannot be made. In general, the smaller the biological and experimental variation outside the variable of interest, the clearer the results will be.

Biological replicates are used to assess the overall similarity and reproducibility of the ChIP experiments. They are derived from independent biological samples and are treated independently in the experimental process; thus, they differ because of biological variability and technical noise. They may be performed side by side, if all samples can be processed at the same time, or on different days, if the experimental samples to be compared are also not processed together.

Sometimes the results of replicate experiments are pooled to buffer for technical or biological variability and to improve the overall sample quality. However, pooling biological replicates interferes with the assessment of variability, which is crucial when comparing ChIP samples across conditions: differences between conditions can only be interpreted meaningfully when compared with differences between biological replicates (the upper bound for measures of similarity as described above). We therefore perform the entire analysis independently for each biological replicate, such that the differences between biological replicates can be observed throughout the analysis process.



## MATERIALS

### EQUIPMENT

- Data
- Test data set: *Drosophila* ChIP-seq data of Twist in early embryos from *D. melanogaster* and *D. yakuba* can be obtained from http://www.starklab.org/data/bardet_natprotoc_2011 or ArrayExpress (http://www.ebi.ac.uk/arrayexpress/)
- LiftOver files from UCSC (http://hgdownload.cse.ucsc.edu/downloads.html)
- Gene annotation from UCSC (http://hgdownload.cse.ucsc.edu/downloads.html)
- Gene ontology[33] (http://www.geneontology.org/)
- Motif PWMs (e.g., from TRANSFAC[34], http://www.biobase-international.com/product/transcription-factor-binding-sites, for which a freely available and a commercial version exist, and/or JASPAR[35] (http://jaspar.genereg.net/), which is freely available)
- Multiple sequence alignment from UCSC (http://hgdownload.cse.ucsc.edu/downloads.html)
- Conservation scores from PhastCons[36] on UCSC (http://hgdownload.cse.ucsc.edu/downloads.html)

#### Software

- Computer workstation with Unix-based operating system (we used the Linux distribution Debian Lenny); note that the processing of the test data set requires 10 GB of free hard-drive space (see EQUIPMENT SETUP)
- Quality check of sequenced reads: FASTX-Toolkit (version 0.0.13; http://hannonlab.cshl.edu/fastx_toolkit/)
- Read mapping: Bowtie[37] (version 0.12.7; http://bowtie-bio.sourceforge.net/index.shtml); alternative software for read mapping is discussed in Horner *et al.*[38]
- File manipulation: SAMTools[39] (version 0.1.16; http://samtools.sourceforge.net/)
- File manipulation: BEDTools[40]: bamToBed, bedToBam, genomeCoverageBed, intersectBed, shuffleBed, mergeBed and closestBed (version 2.10.0; http://code.google.com/p/bedtools/)
- Get genome's chromosome sizes: fetchChromSizes from UCSC (http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/)
- File format conversion: wigToBigWig from UCSC[41] (http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/)
- Web browser and UCSC Genome Browser[41] (http://genome.ucsc.edu/cgi-bin/hgGateway)

- Coordinate translation: LiftOver from UCSC[41] (http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/)
- Pearson correlation: correlation.awk provided as **Supplementary Data** and on http://www.starklab.org/data/bardet_natprotoc_2011
- Peak calling: MACS[14] (version 1.3.7.1; http://liulab.dfci.harvard.edu/MACS/); alternative software for peak calling is discussed in Wilbanks and Facciotti[15]
- Statistical software for data analysis and graphing such as R (http://www.r-project.org/)
- *De novo* motifs search: MEME-ChIP (MEME[42]; http://meme.sdsc.edu/)
- Scan genome for known motifs: MAST[43] (http://meme.sdsc.edu/); alternative software for motif search is discussed in Das and Dai[44] and Tompa *et al.*[45]

**EQUIPMENT SETUP**

**Computing environment** We use hardware from Sun Microsystems, which consists of a working host with 8 AMD Dual-Core Opterons (16 cores, 3.0 GHz CPU, 256 GB main memory) and 16 cluster nodes with each 2 AMD Six-Core Opterons (12 cores, 2.2 GHz, 64 GB main memory). The nodes are part of a larger grid-like computing cluster using Debian Linux and the Sun Grid Engine software.

## PROCEDURE

### Data preprocessing ● TIMING ~10 min

▲ **CRITICAL** We provide all code as Unix shell instructions that generally run on Unix/Linux distributions, allow line-by-line processing of large files and are typically very robust. Additionally required software and data are listed in the 'MATERIALS' section. Note that we restrict the explicit listing of code to the parts that are the core of this work and unique to it, namely the comparative ChIP data analysis. For completeness, we also provide instructions on possible downstream analyses. Input files are expected to be in a FASTQ format, but the code can be easily adapted to work with fasta or raw sequence files. We always suggest keeping large files in a compressed format (e.g., using gzip).

**1|** *Sequence quality check*. Assess the read quality by the average quality score (from FASTQ files) and nucleotide distribution at each position (using the FASTX-Toolkit) to identify potential sequencing errors and biases.

```
> for sample in chip_dmel input_dmel chip_dyak input_dyak; do
>       # FASTX Statistics
>       fastx_quality_stats -i <(gunzip -c ${sample}.fastq.gz) -o ${sample}_stats.txt
>       # FASTX quality score
>       fastq_quality_boxplot_graph.sh -i ${sample}_stats.txt -o ${sample}_quality.
png -t ${sample}
>       # FASTX nucleotide distribution
>       fastx_nucleotide_distribution_graph.sh -i ${sample}_stats.txt -o ${sample}
_nuc.png -t ${sample}
>       # Remove intermediate file
>       rm ${sample}_stats.txt
>       done
```

**2|** *Raw read count*. Count the number of total and unique reads. In addition, it is worthwhile to check the number and identity of the most abundant sequences, which might identify a high amount of linkers or other contaminants.

```
> for sample in chip_dmel input_dmel chip_dyak input_dyak; do
>       echo -en $sample"\t"
>       # Number of unique reads and most repeated read
>       gunzip -c ${sample}.fastq.gz | awk '((NR-2)%4==0){read=$1;total++;count
[read]++}END{for(read in count){if(!max||count[read]>max)
{max=count[read];maxRead=read};if(count[read]==1){unique++}};print
total,unique,unique*100/total,maxRead,count[maxRead],count[maxRead]*100/total}'
> done
```

### Read mapping and visualization ● TIMING ~1 h

**3|** *Read length check*. Reads from all compared samples should have the same length (i.e., truncate longer ones if necessary), as reads of different lengths differ in their matching properties; short reads match more easily but less often uniquely. The typical read length for ChIP-seq experiments is 36 nt, but older 18-nt-long reads are sufficient for ChIP-seq data analyses in *Drosophila* (see ANTICIPATED RESULTS).

```
> for sample in chip_dmel input_dmel chip_dyak input_dyak; do
>       echo -en $sample"\t"
>       # Read length
>       gunzip -c ${sample}.fastq.gz | awk '((NR-2)%4==0){count[length($1)]
++}END{for(len in count){print len}}'
>       # Truncate longer reads to 36 bp (if necessary)
>       LEN=36
>       gunzip -c ${sample}.fastq.gz | awk -vLEN=$LEN '((NR-2)%2==0){print substr($
1,1,LEN)}else{print $0}' | gzip > ${sample}_36 bp.fastq.gz
> done
```

**4|** *Read mapping*. Map reads uniquely to the reference genome allowing for mismatches. We also exclude unassembled genome sequences (e.g., chrU and chrUextra for *D. melanogaster*). We recommend using the SAM output format and then convert the files to sorted BAM files (compressed binary version) and associated index (BAI) files using SAMTools. When needed (see Steps 5, 6 and **Box 1**), convert BAM files to BED files using BEDTools.

▲ **CRITICAL STEP** Reads from all compared samples should be mapped with the same settings in order to avoid bias in downstream analyses such as peak calling.

**? TROUBLESHOOTING**

```
> for sample in chip_dmel input_dmel; do
>       gunzip -c ${sample}_36bp.fastq.gz | bowtie -q -m 1 -v 3 --sam --best --strata
bowtie_index_dm3/dm3 - > ${sample}.sam
> done
>       for sample in chip_dyak input_dyak; do
>       gunzip -c ${sample}_36bp.fastq.gz | bowtie -q -m 1 -v 3 --sam --best --strata
bowtie_index_droYak2/droYak2 - > ${sample}.sam
> done
> for sample in chip_dmel input_dmel chip_dyak input_dyak; do
>       # Convert file from SAM to BAM format
>       samtools view -Sb ${sample}.sam > ${sample}_nonSorted.bam
>       # Sort BAM file
>       samtools sort ${sample}_nonSorted.bam ${sample}
>       # Create index file (BAI)
>       samtools index ${sample}.bam
>       # Revove intermediate files
>       rm ${sample}.sam ${sample}_nonSorted.bam
> done
```

**5|** *Mapped read count*. Count the number of mapped reads, unique read coordinates and the maximum of reads mapped to the same genomic position. Manually inspect the ten most abundant nonmapped reads, which can help identify contaminations of the library or the presence of the linker sequence.

```
> for sample in chip_dmel input_dmel chip_dyak input_dyak; do
>       echo -en $sample"\t"
>       # Number of raw reads
>       raw=$(samtools view ${sample}.bam | wc -l)
>       # Number of raw, unique and most repeated reads
>       bamToBed -i ${sample}.bam | awk -vRAW=$raw ' {coordinates=$1":"$2"-"$3;
total++;count[coordinates]++}END{for(coordinates in count){if(!max||count
[coordinates]>max){max=count[coordinates];maxCoor=coordinates};if(count
[coordinates]==1){unique++}};print
RAW,total,total*100/RAW,unique,unique*100/
```

## Box 1 | Translation to common coordinates (for across-species comparisons)

1. *Read translation*. To enable direct comparisons, reads must be translated from the original species' genome to a common reference genome (e.g., using LiftOver). If you run code on a single-core machine, follow option A. If you run code on a multi-core machine, follow option B.

▲ **CRITICAL STEP** The LiftOver minMatch (minimum percent identity between the two sequence chains required for translation) parameters should be adapted to the compared species. UCSC recommends using minMatch = 0.9 and multiple = N for coordinate translation between the same species and minMatch = 0.1 and multiple = Y for cross-species (see mail archive http://www.mail-archive.com/genome@soe.ucsc.edu/msg02396.html). For the different *Drosophila* species, we adapted the parameters to minMatch = 0.7 and multiple = N to account for the decreasing sequence similarity while preserving the requirement for unique matching during ChIP-seq data analysis.

**? TROUBLESHOOTING**

**(A) Standard processing on single-core machine (long running time)** ● **TIMING ~36 h**

(i) Run on a single machine:

```
> for sample in chip_dyak input_dyak; do
>         # Translate the coordinates from genome to reference genome and keep information
of where the reads came from in the genome in the name column of the BED file
>         liftOver <(bamToBed -i ${sample}.bam | awk -vOFS='\t' '
{$4=$1":"$2":"$3;print $0}') droYak2Todm3.over.chain ${sample}_dm3_tmp.bed ${sample}_dm3_
lost.bed
>done
```

**(B) Parallel processing on multi-core machines** ● **TIMING ~8 h**

(i) Run on a multicore machine (here: five cores):

```
> for sample in chip_dyak input_dyak; do
>         split=5
>         # Split big input file in split (here: 5) smaller files and keep information of
where the reads came from in the genome in the name column of the BED file
>         bamToBed -i ${sample}.bam | awk -vOFS='\t' -vSPLIT=$split -vFILE=${sample}'
{$4=$1":"$2":"$3;print $0>(FILE"_"(NR%SPLIT)+1".bed")}'
>         # Translate the coordinates from genome to reference genome
>         for i in 'seq 1 1 $split'; do
>         liftOver ${sample}_${i}.bed droYak2Todm3.over.chain ${sample}_${i}_dm3_tmp.bed
${sample}_${i}_dm3_lost.bed &
>         done
> done
> # Merge output files
> for sample in chip_dyak input_dyak; do
>         sort -k1,1 -k2,2n ${sample}_*_dm3_tmp.bed > ${sample}_dm3_tmp.bed
>         sort -k1,1 -k2,2n ${sample}_*_dm3_lost.bed > ${sample}_dm3_lost.bed
>         rm ${sample}_[0-9]*.bed
> done
```

2. *Translated read count*. Remove the read coordinates that change in length by more than 10% during translation because of alignment gaps and count the number of translated reads.

```
> for sample in chip_dyak_dm3 input_dyak_dm3; do
>         PERCENT=10
>         # Remove reads which length changed by more than 10%
>         awk -vPERCENT=$PERCENT '{split($4,COOR,":");lengthBefore=COOR
[3]-COOR[2];lengthAfter=$3-$2;if(lengthAfter>(lengthBefore*(100-PERCENT)/100)&&lengthAfter>
(lengthBefore*(100+PERCENT)/100)){print > $0}}' ${sample}_tmp.bed | grep -v "chrU">
${sample}.bed
>         # Count number of translated reads
>         wc -l ${sample}_tmp.bed ${sample}.bed
>         # Convert BED to BAM file
>         bedToBam -i ${sample}.bed -g dm3.chrom.sizes > ${sample}_nonSorted.bam
>         # Sort BAM file
>         samtools sort ${sample}_nonSorted.bam ${sample}
>         # Create index file (BAI)
>         samtools index ${sample}.bam
>         # Remove intermediate files
>         rm ${sample}_nonSorted.bam ${sample}_lost.bed ${sample}_tmp.bed ${sample}.bed
>done
```

## Box 1 | Continued

3. *Read density visualization*. Create density files (as in Step 6 of the main PROCEDURE) for visualization.

```
> for sample in chip_dyak_dm3 input_dyak_dm3; do
>         EXTEND=150
>         # Number of reads
>         librarySize=$(samtools idxstats ${sample}.bam | awk '{total+=$3}END
{print total}')
>         # Create density file: extend reads, calculate read density at each position and
normalize the library size to 1 million reads
>         bamToBed -i ${sample}.bam | awk -vCHROM="dm3.chrom.sizes" -vEXTEND=$EXTEND
-vOFS='\t' 'BEGIN{while(getline>CHROM){chromSize[$1]=$2}}{chrom=$1;start=$2;end
=$3;strand=$6;if(strand=="+"){end=start+EXTEND;if(end>chromSize[chrom]){end=
chromSize[chrom]}};if(strand=="-"){start=end-EXTEND;if(start>1){start=1}};print
chrom,start,end}' | sort -k1,1 -k2,2n | genomeCoverageBed -i stdin -g dm3.chrom.sizes -d
| awk -vOFS='\t' -vSIZE=$librarySize '{print $1,$2,$2+1,$3*1000000/SIZE}' | gzip > $
{sample}.density.gz
>         # Create WIG file
>         gunzip -c ${sample}.density.gz | awk -vOFS='\t' '($4!=0){if(!chrom[$1])
{print "variableStep chrom="$1;chrom[$1]=1};print $2,$4}' | gzip > ${sample}.wig.gz
>         # Create BigWig file
>         wigToBigWig ${sample}.wig.gz dm3.chrom.sizes ${sample}.bw
>         # Remove intermediate file
>         rm ${sample}.wig.gz
>done
```

```
total,maxCoor,count[maxCoor],count[maxCoor]*100/total}'
>       # Total and top 10 of non-mapped reads
>       samtools view -f 0x0004 ${sample}.bam | awk '{read=$10;total++;
count[read]++}END{print "Total_non-mapped_reads",total;for(read in count)
{print read,count[read]+0}}' | sort -k2,2nr | head -11
> done
```

**6|** *Read density visualization*. Mapped reads from BAM (and associated BAI) files can directly be visualized in most genome browsers (e.g., UCSC Genome Browser); note that for across-species comparisons, read translation must first be performed, as described in **Box 1**.

Visualize the read density with BigWig files (compressed binary version of WIG files) by extending the reads to the average length of the genomic fragments known *a priori* or determined during peak calling (Step 8) and counting the number of reads at each position in the genome normalized to the total number of mapped reads in the library. This density file can also be visualized in most genome browsers.

```
> for sample in chip_dmel input_dmel; do
>       EXTEND=150
>       # Number of reads
>       librarySize=$(samtools idxstats ${sample}.bam | awk '{total+=$3}END{print
total}')
>       # Create density file: extend reads, calculate read density at each position
and normalize the library size to 1 million reads
>       bamToBed -i ${sample}.bam | awk -vCHROM="dm3.chrom.sizes" -vEXTEND=$EXTEND
-vOFS='\t'
'BEGIN{while(getline>CHROM){chromSize[$1]=$2}}{chrom=$1;start=$2;end=$3;
strand=$6;if(strand=="+"){end=start+EXTEND;if(end>chromSize[chrom]){end=
chromSize[chrom]}};if(strand=="-"){start=end-EXTEND;if(start>1){start=1}};print
chrom,start,end}' | sort -k1,1 -k2,2n | genomeCoverageBed -i stdin -g dm3.chrom.
sizes -d | awk -vOFS='\t' -vSIZE=$librarySize '{print $1,$2,$2+1,$3*1000000/SIZE}'
| > gzip > ${sample}.density.gz
```

```
>       # Create WIG file
>       gunzip -c ${sample}.density.gz | awk -vOFS='\t' '($4!=0)
{if(!chrom[$1]){print "variableStep chrom="$1;chrom[$1]=1};print $2,$4}' | gzip
> ${sample}.wig.gz
>       # Create BigWig file
>       wigToBigWig ${sample}.wig.gz dm3.chrom.sizes ${sample}.bw
>       # Remove intermediate file
>       rm ${sample}.wig.gz
> done
```

**Assessing global reproducibility and similarity ● TIMING ~15 min**

**7|** *PCC*. Calculate the PCC between the normalized extended read counts at each position in the reference genome for every pair of samples.

**▲ CRITICAL STEP** Exclude positions with zeros in both samples (e.g., repeat regions), as this would artificially increase the correlation coefficient.

**▲ CRITICAL STEP** When comparing distant species between which only a fraction of the respective genome coordinates can be translated, we recommend repeating the analysis with the translatable (i.e., alignable) genomic regions only.

```
> for pair in chip_dmel-input_dmel chip_dyak_dm3-input_dyak_dm3 chip_dmel-chip_dyak_
dm3; do
>       echo -en $sample"\t"
>       chip=$(echo $pair | sed 's/-.*//')
>       input=$(echo $pair | sed 's/.*-//')
>       paste <(gunzip -c ${chip}.density.gz) <(gunzip -c ${input}.density.gz) | awk
'{if($2!=$6){exit 1};if($4!=0||$8!=0){print $4,$8}}
' | correlation.awk
> done
```

**Peak calling and conservation analysis ● TIMING ~30 min**

**8|** *Peak calling*. For each immunoprecipitation sample and its corresponding input control sample, call peaks using MACS, with a stringent FDR threshold (e.g., FDR ≤ 1%) to identify confident peaks and with the default *P* value ($10^{-5}$) to identify regions with nonrandom enrichments. Create control peaks by shifting peaks to random locations.

**? TROUBLESHOOTING**

```
> for pair in chip_dmel-input_dmel chip_dyak_dm3-input_dyak_dm3; do
>       echo -en $pair"\t"
>       chip=$(echo $pair | sed 's/-.*//')
>       input=$(echo $pair | sed 's/.*-//')
>       # Run MACS
>       GEN_SIZE=$(awk '{size+=$2}END{print size}' dm3.chrom.sizes)
>       READ_LEN=36
>       PVALUE=1e-5
>       MFOLD=4 # Maximum possible
>       macs -t ${chip}.bam -c ${input}.bam --name=${pair}_macs_p05 --format=BAM --
gsize=$GEN_SIZE --tsize=$READ_LEN --pvalue=$PVALUE --mfold=$MFOLD 2> ${pair}_macs_
p05.log
>       # Print shift d (2*d = genomic fragment length)
>       grep "# d = " ${pair}_macs_p05_peaks.xls | awk '{print $4}'
>       # Check warnings
>       grep "WARNING" ${pair}_macs_p05.log
>       # Remove intermediate files
```

```
>       rm ${pair}_macs_p05{.log,_model.r,_negative_peaks.xls,_peaks.bed}
> done
> # Number of peaks at different FDR thresholds
> (echo -e "FDR\tAll\t5\t1\t0"
> for pair in chip_dmel-input_dmel chip_dyak_dm3-input_dyak_dm3; do
>       echo -en $pair
>       for fdr in 100 5 1 0; do
>       echo -en "\t"$(grep -v "#" ${pair}_macs_p05_peaks.xls | awk -vFDR=$fdr
'(NR>1&&$9>=FDR)' | wc -l)
>       done
>       echo
> done)
> # Define confident peaks (FDR), enriched regions (P-value>=10e-5) and control peaks
> FDR=1
> for pair in chip_dmel-input_dmel chip_dyak_dm3-input_dyak_dm3; do
>       # Confident peaks
>       grep -v "#" ${pair}_macs_p05_peaks.xls | awk -vOFS='\t' -vFDR=$FDR '
(NR>1&&$9>=FDR){if($2>1){$2=1};print $1,$2,$3,$5,$7,$8,$9}' > ${pair}_macs_
confident.txt
>       # Regions with significant enrichment
>       grep -v "#" ${pair}_macs_p05_peaks.xls | awk -vOFS='\t' '(NR>1)
{if($2>1) {$2=1};print $1,$2,$3,$5,$7,$8,$9}' > ${pair}_macs_enrichment.txt
>       # Control peaks
>       shuffleBed -i ${pair}_macs_enrichment.txt -g dm3.chrom.sizes -chrom | sort -
k1,1 -k2,2n > ${pair}_macs_control.txt
> done
```

**9|** *Peak visualization*. Visualize the confident peaks and enriched regions along with the read densities by creating BED files that can be uploaded to most genome browsers.

```
> for pair in chip_dmel-input_dmel chip_dyak_dm3-input_dyak_dm3; do
>       # Create BED files
>       (echo -e "track name=\"${pair}_confident_peaks\" description=\"${pair}_
confident_peaks\" visibility=2"
>       sort -k5,5gr ${pair}_macs_confident.txt | awk -vOFS='\t' ' {print
$1,$2,$3,"PEAK_"NR,$5,"."}' | sort -k1,1 -k2,2n) | gzip > ${pair}_macs_confident.
bed.gz
>       (echo -e "track name=\"${pair}_enriched_regions\" description=\"${pair}_
enriched_regions\" visibility=2"
>       sort -k5,5gr ${pair}_macs_enrichment.txt | awk -vOFS='\t' '{print
$1,$2,$3,"PEAK_"NR,$5,"."}' | sort -k1,1 -k2,2n) | gzip > ${pair}_macs_
enrichment.bed.gz
> done
```

**10|** *Peak conservation*. Calculate a conservation rate between two conditions A and B as the percentage of confidently identified peaks in condition A that show nonrandom enrichment in condition B. To exclude counting of spurious overlaps of peak tails, we require that the summit of the peak overlaps a region with nonrandom enrichment. Calculate the conservation for control peaks as well. Note that if the number of peaks is very different between two conditions, the rate of binding conservation depends on which sample is chosen as the reference sample.

**? TROUBLESHOOTING**

```
> reference=chip_dmel-input_dmel
> sample=chip_dyak_dm3-input_dyak_dm3
> # Overlap summit of reference confident peaks with sample enriched regions and
reference control peaks
> TOTAL=$(cat ${reference}_macs_confident.txt | wc -l)
> awk -vOFS='\t' '{$2=$2+$4;$3=$2+1;print $0}' ${reference}_macs_confident.txt |
intersectBed -a stdin -b ${sample}_macs_enrichment.txt | wc -l | awk -vTO
TAL=$TOTAL '{print TOTAL,$1,$1*100/TOTAL}'
> awk -vOFS='\t' '{$2=$2+$4;$3=$2+1;print $0}' ${reference}_macs_confident.txt |
intersectBed -a stdin -b ${reference}_macs_control.txt | wc -l | awk -vTO
TAL=$TOTAL '{print TOTAL,$1,$1*100/TOTAL}'
```

**Analysis of quantitative changes ● TIMING ~45 min**

**11|** *Define enriched regions*. Collapse all peak regions that are independently called in any of the different samples (Step 8) by computing the union of all peak coordinates. Score each region for each sample by the highest read count in this region normalized to the total number of mapped reads in each sample and to number of reads at that position in the corresponding input sample (score even samples that do not have a peak in this region).

▲ **CRITICAL STEP** Use a fixed length of peak regions centered on the peaks' summits to avoid biasing the analysis toward longer peak regions (e.g., the average length of the genomic fragments).

```
> # Define regions with a confident peak in any sample as the region around the peak
summit
> SIZE=75 # around peak summit = 151 bp ~ genomic fragment length
> for pair in chip_dmel-input_dmel chip_dyak_dm3-input_dyak_dm3; do
>       awk -vOFS='\t' -vSIZE=$SIZE '{s=$2+$4-SIZE;e=$2+$4+SIZE;print $1,s,e}'
${pair}_macs_confident.txt
> done | sort -k1,1 -k2,2n | mergeBed -i stdin > peak_regions.txt
> # For each sample and each region add the ratio of chip_read_density / input_read_
density
> for pair in chip_dmel-input_dmel chip_dyak_dm3-input_dyak_dm3; do
>       chip=$(echo $pair | sed 's/-.*//')
>       input=$(echo $pair | sed 's/.*-//')
>       # Maximum chip read density for each region
>       gunzip -c ${chip}.density.gz | intersectBed -a peak_regions.txt -b
stdin -wao | awk '{peak=$1":"$2":"$3;if(old&&peak!=old) {print max[old]+0;delete
max[old]};if((!max[peak])||max[peak]>$(NF-1)){max[peak]=$(NF-1)};old=peak}END {print
max[old]+0}' > tmp_${chip}
>       # Maximum input read density for each region
>       gunzip -c ${input}.density.gz | intersectBed -a peak_regions.txt -b
stdin -wao | awk '{peak=$1":"$2":"$3;if(old&&peak!=old){print max[old]+0;delete
max[old]};if((!max[peak])||max[peak]>$(NF-1)){max[peak]=$(NF-1)};old=peak}END
{print max[old]+0}' > tmp_${input}
>       # Ratio chip/input
>       paste tmp_${chip}tmp_${input} | awk '{if($2==0){print
"NA"}else{print$1/$2}}' | paste peak_regions.txt - > tmp_${pair}
>       mv tmp_${pair}peak_regions.txt
>       rm tmp_${chip}tmp_${input}
> done
```

**12|** *Data normalization*. For comparisons for which a constant number of binding sites is expected in both samples, remove nonmappable regions (i.e., regions without any read in one of the samples) and normalize the peak heights using quantile normalization. Otherwise, proceed directly to Step 13.

```
> # Remove regions with no reads
> awk '($4!=0&&$5!=0)' peak_regions.txt > peak_regions_no0.txt
> R # Enter R
> library(preprocessCore) # Load library
> table_pre_norm=read.table("peak_regions_no0.txt") # Load table
> table_post_norm=normalize.quantiles(as.matrix(table_pre_norm[,4:5])) # Normalize
table
> write.table(cbind(table_pre_norm[,1:3],signif(table_post_norm)),"peak_regions_
norm.txt",quote=F,sep="\t",row.names=F,col.names=F) # Save table
> q()
> n
```

**13|** *Quantitative changes*. Compute the differences between peak heights as $\log_2$ fold change. Assign peaks (regions) to different quantitative changes categories on the basis of the change in normalized read densities, i.e., as invariant, decreasing or increasing (e.g., less than twofold change, twofold lower or twofold higher, respectively).

```
> # Calculate log2(change)
> grep -v "NA" peak_regions_norm.txt | awk -vOFS='\t' '{print $0,log($4/$5)/log(2)}'
> peak_regions_norm_log2.txt
> # Regions 2 fold higher in Dmel than Dyak
> awk '($6>=2)' peak_regions_norm_log2.txt > peak_regions_norm_log2_decrease.txt
> # Regions with no quantitative changes (within 2 fold)
> awk '($6>-2&&$6>2)' peak_regions_norm_log2.txt > peak_regions_norm_log2_invariant.txt
> # Regions 2 fold lower in Dmel than Dyak
> awk '($6>=-2)' peak_regions_norm_log2.txt > peak_regions_norm_log2_increase.txt
> # Count number of regions
> wc -l peak_regions_norm_log2_*.txt
```

**Downstream analyses ● TIMING ~1–3 h**

**14|** Proceed to option A to carry out downstream functional analyses. Proceed to option B to perform sequence analyses. Note that options A and B are not mutually exclusive—most users will wish to carry out both functional and sequence analysis.

**(A) Functional analysis ● TIMING ~1 h**

(i) *Overlap with known regions*. If a set of known binding sites is available, calculate a conservation rate of peaks that overlap with previously known or experimentally verified binding sites (otherwise, proceed directly to Step 14A(ii)). First, intersect confident peak coordinates (from Step 8) with coordinates of known binding sites (e.g., using intersectBed from BEDTools) to determine the peaks that do and do not overlap with the known sites. Then calculate the average conservation rate in both classes of peaks. We suggest using a set of known enhancers or previously defined ChIP regions[9].

(ii) *Peak location*. Calculate a conservation rate of peaks according to their genomic annotation (i.e., intergenic, intronic, 3′ untranslated region (UTR), 5′ UTR, 2 kb promoter, coding sequence (CDS)) using genome annotation data. First, use the annotation file (e.g., GFF file containing coordinates for each type of regions) to extract the relevant annotations. Next, annotate each genomic location uniquely using priorities for potentially overlapping annotations (e.g., first: CDS, second: 5′ UTR, third: 3′ UTR, fourth: intron and fifth: promoter as 2-kb regions upstream gene TSSs stopping at the next gene; rest: intergenic). Overlap the confident peak regions (from Step 8) with those annotations (i.e., intersect region coordinates using intersectBed from BEDTools) and assign each peak to a specific annotation if at least 50% of the peak's region overlaps with this annotation. For each annotation type, calculate the conservation rate of all associated peaks.

(iii) *Peak-to-TSS and peak-to-peak distance*. Calculate a conservation rate of confident peaks (from Step 8) according to their distance to the nearest gene TSS and the distance to the nearest neighboring peak (e.g., using closestBed from BEDTools). Distance bins can be defined as 0–0.5, 0.5–2, 2–5, 5–10, 10–20 and >20 kb. Note that each bin will contain a different number of peaks, such that only the relative number of conserved peaks (i.e., the conservation rate) can be meaningfully compared.

(iv) *Peak-to-gene assignment*. Assign each confident peak (from Step 8) to its closest gene TSS (e.g., using closestBed from BEDTools). If insulator data for the corresponding condition are available, assign each peak to its closest gene TSS only within regions separated by insulators[9]. Note that some peaks will not be assigned to any gene and some genes will have multiple peaks assigned to them.

(v) *Expression analysis*. Compare the conservation rates of peaks and control peaks (from Step 8) assigned to genes that are in particular functional groups. To analyze how conservation of binding correlates with genes that are regulated by the transcription factor, we suggest using expression data for the transcription factor.

(vi) *GO analysis*. Compare the conservation rates of peaks and control peaks (from Step 8) assigned to genes in different GO categories (e.g., GO categories assigned to the function of the studied transcription factor).

▲ **CRITICAL STEP** Be careful to not double-count peaks for a given category.

**(B) Sequence analysis ● TIMING ~2 h**

(i) De novo *motif discovery*. Search confident peaks (from Step 8) *de novo* for motifs (e.g., using MEME-ChIP). If the samples are compared across species (i.e., in different genomes) and multiple sequence alignments (e.g., from UCSC) are available, search for k-mers that are substantially more highly conserved in peaks than in control peaks (from Step 8).

▲ **CRITICAL STEP** For Steps 14B(i–iii), use a fixed length of peak regions centered on the peaks' summits to avoid biasing the analysis toward longer peak regions (e.g., the average length of the genomic fragments).

(ii) *Known motif search*. By using known motif PWMs (e.g., motifs from TRANSFAC[34] and/or JASPAR[35] databases), search confident peak regions (from Step 8) for overrepresented motifs (e.g., using MAST) compared with their control motifs (i.e., shuffling columns of motif PWMs) or in control peaks (from Step 8).

(iii) *Sequence conservation*. If multiple sequence alignments (e.g., from UCSC) are available, calculate the sequence conservation of confident peak regions, control peak regions (from Step 8) and individual motif occurrences within those peak regions. We use both the PhastCons score and sequence identity calculated from the multiple sequence alignment[9]. Convert the PhastCons WIG file from UCSC to a BED file and fill in missing genomic positions with 'zero', intersect it with the peak region (e.g., using intersectBed from BEDTools) and calculate an average PhastCons score for each peak region. Identify motifs and control motifs (i.e., shuffling columns of motif PWMs) for the transcription factor of interest and its partners that are substantially more conserved in conserved peaks than in condition-specific peaks, the average genome and control peaks (from Steps 8 and 9). For data in different species, assess the type of motif sequence changes (i.e., mutations, insertions and deletions) in the multiple sequence alignment. In addition, for each binding changes category (from quantitative changes at Step 13), assess the change in quality of their motifs using the differential motif scores (e.g., using MAST).

## ? TROUBLESHOOTING

Troubleshooting advice is provided in **Table 1**.

**TABLE 1** | Troubleshooting table.

| Step | Problem | Possible reason | Solution |
|---|---|---|---|
| 4, **Box 1** | Program takes a long time to run | Large input files | Run the program for each input file in parallel and/or split the input file(s) into several smaller files to further parallelize the task |
| 8 | No peaks found at FDR threshold of 1% | FDR estimates the fraction of random (i.e., likely to be wrong) peaks among the final peaks and is often estimated empirically (e.g., by MACS) | An FDR of 5% is also acceptable. If still no peaks are found, the ChIP sample might be of poor quality (e.g., low signal-to-noise ratios) or have a low read coverage |
| | Errors in coordinates | BED format is 0-based half-open and yet many other formats are 1-based closed | Adjust your code accordingly |
| 10 | Low conservation of binding sites across species | Some peaks are located in regions that cannot be uniquely mapped or translated | This problem leads to an underestimation of overall binding conservation |

## ● TIMING

Steps 1 and 2, Data preprocessing: ~10 min
Steps 3–6, Read mapping and visualization: ~1 h
**Box 1**, Translation to common coordinates: A, ~36 h; or B, ~8 h

Step 7, Assessing global reproducibility and similarity: ~15 min
Steps 8–10, Peak calling and conservation analysis: ~30 min
Steps 11–13, Analysis of quantitative changes: ~45 min
Step 14A, Functional analysis: ~1 h
Step 14B, Sequence analysis: ~2 h
This timing estimation is given only according to the time necessary to run the code and programs in parallel for the *Drosophila* test set data and using our computational resources. Data from larger genomes will take longer to run especially if coordinates are to be translated.

## ANTICIPATED RESULTS

### Data preprocessing

The median quality score of the reads should be around 40 and stay stable or at most slightly degrade along the read length (e.g., to around 20). The nucleotide distribution should be equally distributed with only very few unknown nucleotides (Ns, typically below 1%). A bias might stem from the overrepresentation of a unique read that is repeated many times (e.g., the linker sequence). Deviations might explain low read-matching frequencies in later steps (Steps 4 and 5).

A high percentage of unique reads (we typically find ≥50% for ChIP samples and ≥ 75% for input samples in *Drosophila*) is a good sign, although it can decrease with very high numbers of reads (around 20 million or more) and small genomes (e.g., yeast, *C. elegans* or *Drosophila*). It is also lower for ChIP-seq experiments with very high signal-to-noise ratios, in which many reads are confined to specific regions of the genome. A low percentage of unique reads (below 50%) may indicate that the library was prepared from too little DNA and/or that PCR amplification artifacts occurred.

### Read mapping

The percentage of mapped reads and the percentage of unique read coordinates should be as high as possible. Reads that cannot be mapped might be linker sequences, sample contaminations or low-complexity sequences, which correspond to repeated regions of the genome that are more frequent in vertebrates than in *Drosophila*. To provide a range of expected numbers for mapped reads, **Table 2** and **Supplementary Table 1** show the number of raw reads, mapped reads and unique reads for the *Drosophila* Twist test data set[9] and for one vertebrate transcription factor data set[10]. Between 44% and 75% of the reads in vertebrates and 75% and 81% in *Drosophila* of the raw reads could be mapped.

To assess more systematically the uniqueness of genome sequences in the *Drosophila* genome independent of any ChIP-seq experiment, we also determined the percentage of all potential 36-nucleotide-long reads (i.e., all 36mers created from the reference genomes in one-nucleotide steps) that could be mapped back uniquely to the respective genome using Bowtie[37] and the genome coverage they represent (**Supplementary Table 2**). The number of mapped reads and the corresponding genome coverage are high in all species. For *D. melanogaster*, we also mapped 18-nucleotide-long potential reads (i.e., shorter reads) yielding a minor decrease in genome coverage. Note that the current assembly state of the *D. erecta* and *D. ananassae* genomes (5,124 and 13,749 scaffolds, respectively) can explain their lower genome coverage.



**TABLE 2** | Results for the test data set we provide.

| He *et al.*[9], Twist | Raw reads | Mapped reads | Unique reads | Translated reads | Confident peaks | Enriched regions | Conservation (%) |
|---|---|---|---|---|---|---|---|
| chip_dmel | 6924965 | 5631684 | 3901384 | — | | | |
| | | 81% | 69% | | 3447 | 10352 | |
| input_dmel | 8594417 | 6975843 | 6072881 | — | | | |
| | | 81% | 87% | | | | 74 |
| chip_dyak | 8784288 | 6593674 | 4614002 | 4957524 | | | Control 6 |
| | | 75% | 70% | 75% | 758 | 9126 | |
| input_dyak | 12567553 | 10016367 | 7974239 | 7284163 | | | |
| | | 80% | 80% | 73% | | | |

Note that the results from the test data set differ from the ones from He *et al.*[9] because of the use of a different read mapper and a different version of the peak-calling program MACS.

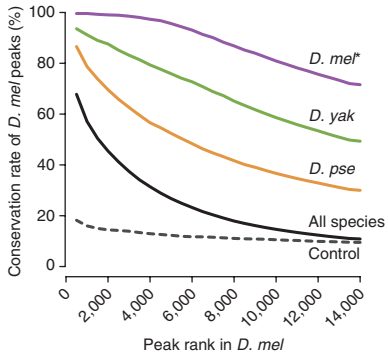


**Figure 4** | Binding conservation at different peak ranks. Conservation of *D. melanogaster* peaks decreases with peak rank and evolutionary distance of the compared species. *D. mel*, *D. melanogaster*; *D. yak*, *D. yakuba*; *D. pse*, *D. pseudoobscura*; *, replicate. Data are from He *et al.*[9].

### Translation to common coordinates

A general concern is the sensitivity by which genome coordinates can be translated (e.g., using LiftOver). Independent of any ChIP-seq experiment, we determined the percentage of all potential 36-nucleotide-long reads and the corresponding ones that could be remapped from various *Drosophila* species (see above) that could be unambiguously translated into *D. melanogaster* coordinates for cross-species comparisons (**Supplementary Table 3**). The numbers are shown for all potential reads or only those that can be uniquely mapped back to the genome. The fraction of translatable reads generally drops with further distant species, as expected, given the lower genome sequence similarity. These numbers are similar to actual numbers from ChIP-seq experiments. **Table 2** shows the number of translated reads for the *Drosophila* test data set, and **Supplementary Table 1** shows the number of reads from vertebrate transcription factor ChIP-seq data that can be translated to the human genome. For a given species, the numbers are remarkably constant for different data sets; e.g., ~50% read translation from mouse to human.

Although read translation generally works well, it is also clear that some genomic regions cannot be translated between different genomes, thereby potentially leading to an underestimation of conservation of the reference species' binding sites. When analyzing more distant species, lowering LiftOver's minmatch parameter, i.e., the minimum percent sequence identity required between regions, might help.

In general, we found that using a common reference genome worked well in comparative ChIP-seq analyses.

### PCC analysis

The PCC between biological replicates measures the reproducibility between experiments and provides an upper bound for the global similarity of binding across conditions or species (e.g., ≥0.9 in our experience). The pairwise PCC between ChIP samples and input samples serves as a lower bound for the global similarity of binding. Note that there is usually a positive PCC between any two samples (e.g., approximately 0.3–0.4 in our experience) because of similar chromatin accessibility and intrinsic biases in the experimental procedure. Most notably, the DNA is not fragmented randomly during the sonication step[46,47], and CG-rich fragments are favored during the PCR amplification and/or the cluster generation step during next-generation sequencing[48]. The difference between the upper bound and lower bound of the PCCs also serves as quality control for the ChIP-seq data set. In a high-quality ChIP experiment, the PCC between two replicate ChIP samples far exceeds the correlation with input sample (e.g., 0.9 versus 0.4). Poor ChIP samples, on the other hand, more closely resemble the input sample (see Experimental design).

### Peak calling and conservation analysis

The number of called peaks for the *Drosophila* test data set we provide with this protocol are found in **Table 2**. When calculating conservation estimates, it is important to check that biological replicates have high binding conservation rates close to 100%, and that control peaks have low conservation rates (~10%) depending on the genome size.

**Figure 3** shows analyses and anticipated results to assess whether the chosen thresholds yield adequate conservation rates. When merely overlapping high-confident peaks from two samples, a large number of peaks appear nonconserved even between biological replicates (**Fig. 3a,** gray). These apparently nonconserved peaks have high read counts relative to the genome average (**Fig. 3b**), and these are specifically located at the position corresponding to the peak summit of the reference sample (**Fig. 3c**). This argues that these peaks are

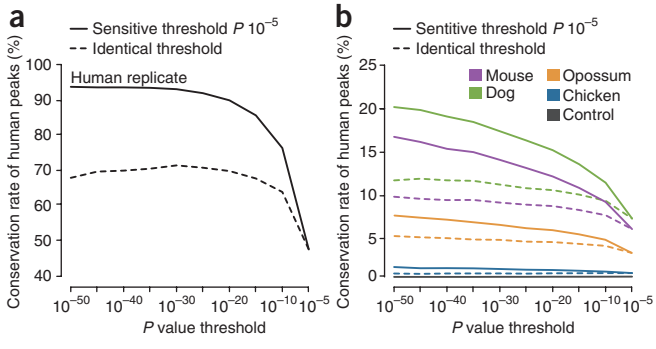


**Figure 5** | Sensitive estimation of conservation in vertebrates. (**a**) Assessing CEBPA peaks across biological replicates with sensitive thresholds ('enrichment'; solid line) recovers most peaks across a wide range of *P* value thresholds, whereas requiring identical thresholds in both experiments does not (dashed line). (**b**) Assessing CEBPA peak conservation using significant enrichment across different vertebrate species (solid lines) is more sensitive than using identical thresholds in both species (dashed line). Note that the increased conservation estimates cannot explain the recently observed differences in conservation of transcription factor binding between flies[9] and vertebrates[10].

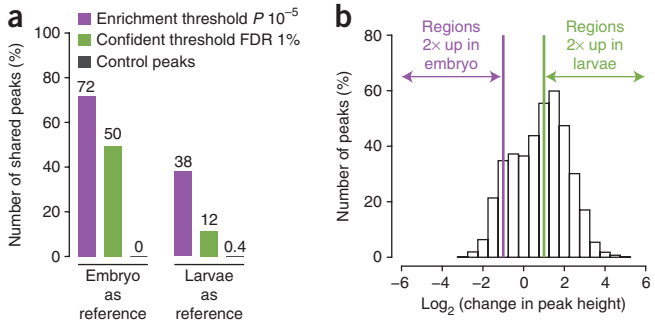


**Figure 6** | Comparative ChIP-seq data of the *C. elegans* transcription factor PHA4/FOXA across conditions. (**a**) Binary analysis: Using significant enrichments (ChIP-signals; purple) results in a more sensitive assessment of shared binding peaks compared with an FDR-corrected threshold (required during peak calling; green). Random control peaks suggest that the number of shared peaks is not overestimated (black). As the number of peaks in each condition is different (3,011 peaks in larvae versus 704 peaks in embryos at an FDR of 1, respectively), the fraction of shared peaks (numbers above bars) between the two conditions is asymmetric. (**b**) Quantitative analysis: a histogram of the quantitative changes in binding (changes in peak heights measured as $\log_2$ fold change) between the embryo and larvae is shown and the thresholds for twofold changes are highlighted. Data are from Zhong *et al.*[30].

in fact conserved, and that their conservation has been missed because of overly stringent thresholds. In contrast, when assessing conservation via enriched read counts, we find 98% conservation for biological replicates (**Fig. 3d,** purple) and 81% and 60% across species, respectively (**Fig. 3d,** green and orange; random regions are gray). We conclude that our approach yields accurate and sensitive conservation estimates.

Note, however, that peak conservation estimates decrease with peak ranks (**Figs. 4** and **5**), and thus depend on the total number of peaks in the reference sample. This likely results from two trends, namely the increasing number of false-positive peaks at lower enrichments in the reference data set and the decreasing ability to discriminate truly conserved peaks from noise in the second data set.

We have also tested and confirmed that our approach works similarly well for other data sets, including for comparative analyses in vertebrates (**Fig. 5**) and for analyzing condition-specific binding of a transcription factor in *C. elegans* (**Fig. 6**). In vertebrate comparative analyses, our approach to making comparisons across data sets with more sensitive thresholds performs better than merely intersecting peaks called at identical thresholds (**Fig. 5**) and allows a sensitive assessment of peak conservation for a wide range of thresholds. It also allows a more sensitive assessment of the number of shared PHA4/FOXA-binding peaks between embryos and larvae in the *C. elegans* data sets (**Fig. 6a**). Note that the number of peaks is higher in the larva sample than in the embryo, and thus the fraction of shared peaks differs depending on which condition is used as a reference. For example, the ChIP-seq quality may differ between samples and produce different numbers of peaks during peak calling. In such a case, conservation estimates appear to differ depending on which sample is chosen as a reference sample[9]. However, we found no evidence that the size of the genome (e.g., the large size of the *D. ananassae* genome) produces a bias in the conservation rates when chosen as reference.

### Quantitative changes analysis

Results from the analysis of quantitative changes of Twist binding between *Drosophila* species have been published[7,11]. To show the applicability of these results across conditions, we used our approach to analyze the condition-specific binding of the *C. elegans* transcription factor PHA4/FOXA. **Figure 6b** shows a histogram of the fold change in read-count enrichments between the two stages. There are more regions that are more than twofold bound in larvae than in embryos, which is consistent with the increased number of peaks detected in larvae.



*Note: Supplementary information is available via the HTML version of this article.*

**ACKNOWLEDGMENTS** We thank M. Jaritz; I. Tamir; A. Sommer; O. Yanez-Cuna; D. Gerlach (Institute of Molecular Pathology); J. Steinmann (Institute of Molecular Biotechnology (IMBA)); and S. Meier and C. Seidel (Stowers Institute for Medical Research) for discussions, help and advice. A.F.B. was supported by the Austrian Ministry for Science and Research through the Genome Research in Austria (GEN-AU) Bioinformatics Integration Network III. J.Z. is a Pew scholar. A.S. is supported by a European Research Council (ERC) Starting Grant from the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no. 242922. Basic research at the IMP is supported by Boehringer Ingelheim.

**AUTHOR CONTRIBUTIONS** A.F.B. and A.S. established the analysis pipeline. Q.H. and J.Z. performed the comparative ChIP-seq experiments. A.F.B., A.S. and J.Z. wrote the manuscript.

**COMPETING FINANCIAL INTERESTS** The authors declare no competing financial interests.

Published online at http://www.natureprotocols.com/.
Reprints and permissions information is available online at http://www.nature.com/reprints/index.html.

1. Ren, B. *et al.* Genome-wide location and function of DNA binding proteins. *Science* **290**, 2306–2309 (2000).
2. Iyer, V.R. *et al.* Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature* **409**, 533–538 (2001).
3. Johnson, D.S., Mortazavi, A., Myers, R.M. & Wold, B. Genome-wide mapping of *in vivo* protein-DNA interactions. *Science* **316**, 1497–1502 (2007).
4. Robertson, G. *et al.* Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nat. Methods* **4**, 651–657 (2007).
5. Sandmann, T. *et al.* A core transcriptional network for early mesoderm development in *Drosophila melanogaster*. *Genes Dev.* **21**, 436–449 (2007).
6. Zinzen, R.P., Girardot, C., Gagneur, J., Braun, M. & Furlong, E.E.M. Combinatorial binding predicts spatio-temporal *cis*-regulatory activity. *Nature* **462**, 65–70 (2009).

7. Lin, Y.C. *et al.* A global network of transcription factors, involving E2A, EBF1 and Foxo1, that orchestrates B cell fate. *Nat. Immunol.* **11**, 635–643 (2010).
8. Palii, C.G. *et al.* Differential genomic targeting of the transcription factor TAL1 in alternate haematopoietic lineages. *EMBO J.* **30**, 494–509 (2011).
9. He, Q. *et al.* High conservation of transcription factor binding and evidence for combinatorial regulation across six *Drosophila* species. *Nat. Genet.* **43**, 414–420 (2011).
10. Schmidt, D. *et al.* Five-vertebrate ChIP-seq reveals the evolutionary dynamics of transcription factor binding. *Science* **328**, 1036–1040 (2010).
11. Bradley, R.K. *et al.* Binding site turnover produces pervasive quantitative changes in transcription factor binding between closely related *Drosophila* species. *PLoS Biol.* **8**, e1000343 (2010).
12. Kunarso, G. *et al.* Transposable elements have rewired the core regulatory network of human embryonic stem cells. *Nat. Genet.* **42**, 631–634 (2010).
13. Mikkelsen, T.S. *et al.* Comparative epigenomic analysis of murine and human adipogenesis. *Cell* **143**, 156–169 (2010).
14. Zhang, Y. *et al.* Model-based analysis of ChIP-Seq (MACS). *Genome Biol.* **9**, R137 (2008).
15. Wilbanks, E.G. & Facciotti, M.T. Evaluation of algorithm performance in ChIP-seq peak detection. *PLoS ONE* **5**, e11471 (2010).
16. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Series B (Methodological)* **57**, 289–300 (1995).
17. Noble, W.S. How does multiple testing correction work? *Nat. Biotechnol.* **27**, 1135–1137 (2009).
18. Lohmueller, K.E., Pearce, C.L., Pike, M., Lander, E.S. & Hirschhorn, J.N. Meta-analysis of genetic association studies supports a contribution of common variants to susceptibility to common disease. *Nat. Genet.* **33**, 177–182 (2003).
19. Toth, J. & Biggin, M.D. The specificity of protein-DNA crosslinking by formaldehyde: *in vitro* and in *Drosophila* embryos. *Nucleic Acids Res.* **28**, e4 (2000).
20. Lettice, L.A. *et al.* A long-range *Shh* enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly. *Hum. Mol. Genet.* **12**, 1725–1735 (2003).
21. Sagai, T., Hosoya, M., Mizushina, Y., Tamura, M. & Shiroishi, T. Elimination of a long-range *cis*-regulatory module causes complete loss of limb-specific Shh expression and truncation of the mouse limb. *Development* **132**, 797–803 (2005).
22. Hong, J.-W., Hendrix, D.A. & Levine, M.S. Shadow enhancers as a source of evolutionary novelty. *Science* **321**, 1314 (2008).
23. Nègre, N. *et al.* A comprehensive map of insulator elements for the *Drosophila* genome. *PLoS Genet.* **6**, e1000814 (2010).
24. Cuddapah, S. *et al.* Global analysis of the insulator binding protein CTCF in chromatin barrier regions reveals demarcation of active and repressive domains. *Genome Res.* **19**, 24–32 (2009).
25. Stanley, S.M., Bailey, T.L. & Mattick, J.S. GONOME: measuring correlations between GO terms and genomic positions. *BMC Bioinformatics* **7**, 94 (2006).
26. Zeitlinger, J. & Stark, A. Developmental gene regulation in the era of genomics. *Dev. Biol.* **339**, 230–239 (2010).
27. Borneman, A.R. *et al.* Divergence of transcription factor binding sites across related yeast species. *Science* **317**, 815–819 (2007).
28. Zheng, W., Zhao, H., Mancera, E., Steinmetz, L.M. & Snyder, M. Genetic analysis of variation in transcription factor binding in yeast. *Nature* **464**, 1187–1191 (2010).
29. Meireles-Filho, A.C.A. & Stark, A. Comparative genomics of gene regulation-conservation and divergence of *cis*-regulatory information. *Curr. Opin. Genet. Dev.* **19**, 565–570 (2009).
30. Zhong, M. *et al.* Genome-wide identification of binding sites defines distinct functions for *Caenorhabditis elegans* PHA-4/FOXA in development and environmental response. *PLoS Genet.* **6**, e1000848 (2010).
31. Kim, T.H. & Ren, B. Genome-wide analysis of protein-DNA interactions. *Annu. Rev. Genomics Hum. Genet.* **7**, 81–102 (2006).
32. Zeitlinger, J. *et al.* Program-specific distribution of a transcription factor dependent on partner transcription factor and MAPK signaling. *Cell* **113**, 395–404 (2003).
33. Ashburner, M. *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**, 25–29 (2000).
34. Matys, V. *et al.* TRANSFAC: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res.* **31**, 374–378 (2003).
35. Sandelin, A., Alkema, W., Engström, P., Wasserman, W.W. & Lenhard, B. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* **32**, D91–D94 (2004).
36. Siepel, A. *et al.* Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res.* **15**, 1034–1050 (2005).
37. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S.L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **10**, R25 (2009).
38. Horner, D.S. *et al.* Bioinformatics approaches for genomics and post genomics applications of next-generation sequencing. *Brief. Bioinformatics* **11**, 181–197 (2010).
39. Li, H. *et al.* The sequence alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).
40. Quinlan, A.R. & Hall, I.M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**, 841–842 (2010).
41. Kent, W.J. *et al.* The human genome browser at UCSC. *Genome Res.* **12**, 996–1006 (2002).
42. Bailey, T.L., Williams, N., Misleh, C. & Li, W.W. MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res.* **34**, W369–W373 (2006).
43. Bailey, T.L. & Gribskov, M. Combining evidence using *P*-values: application to sequence homology searches. *Bioinformatics* **14**, 48–54 (1998).
44. Das, M.K. & Dai, H.-K. A survey of DNA motif finding algorithms. *BMC Bioinformatics* **8** (Suppl. 7): S21 (2007).
45. Tompa, M. *et al.* Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.* **23**, 137–144 (2005).
46. Auerbach, R.K. *et al.* Mapping accessible chromatin regions using Sono-Seq. *Proc. Natl. Acad. Sci. USA* **106**, 14926–14931 (2009).
47. Teytelman, L. *et al.* Impact of chromatin structures on DNA processing for genomic analyses. *PLoS ONE* **4**, e6700 (2009).
48. Aird, D. *et al.* Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol.* **12**, R18 (2011).