**Kent Lab MNase-seq and DNaseI-seq (CPSA) – Raw sequence to PartN files.**

**Ref:** Kent N.A., Adams S., Moorhouse, A. & Paszkiewicz, K. (2011) Chromatin particle spectrum analysis: a method for comparative chromatin structure analysis using paired-end mode next-generation DNA sequencing. *Nucleic Acids Res.* **39**:e26.

Perl scripts should each be given their own directories. The scripts are written to take their relevant input files from that core directory and will generate within, a time-stamped output directory each time they are run.

***A. THE ROSSI CHIP-EXO-SEQ DATA.***

ChIP-exo-seq genomic binding locations (in .bed format) for your factors can be found at:

<https://github.com/CEGRcode/2021-Rossi_Nature/tree/master/04_ChExMix_Peaks>

To download each file, click on it, select Raw Download. Move file somewhere organised.

Note: The Pugh lab (Source of the Rossi paper) are notorious in the field for making their published data available *but* hard to use! The .bed files are no exception – although the paper states that data is quoted relative to the “SacCer3” reference genome, the chromosome designations given in the .bed files use the nomenclature from the previous “SacCer2” reference – e.g., data for chromosome 4 is tagged “chr4” instead of “chrIV”. As you may have noticed computers are literal about their inputs, and this apparently trivial change fecks-up all sorts of things down the line. Nick has written a simple Perl script which changes all the chromosome designations to the right format.

Move your set of downloaded .bed files into the directory you created which contains the script Rossi\_bed\_chr\_changer\_v22\_1.plx. From within that directory, issue the commands:

>perl Rossi\_bed\_chr\_changer\_v22\_1.plx

The script will create a time-stamped output directory containing .bed files with the correct chr designation – these should be used for the subsequent processing.

***B. THE DNA SEQUENCE.***

Typically for each experiment, MNase/DNase-resistant DNA fragments from about 108 cells will have been sequenced in paired-end mode on an Illumina NGS platform to yield a minimum of 1 read-pair per genomic base pair (e.g., >=12M read-pairs per budding yeast genome). Raw sequences for this type of analysis are now typically 50-100nt long (depending on the Illumina kit type). However, in the 2023 BI3001 project, we are using some really old, partially unpublished data, created at the dawn of NGS – these sequence reads were produced by an Illumina GAIIx (which was powered by steam, strong coffee and exceptionally bad language) and yielded reads of only 36bp.

Sequences in .fastq format are in a .zip archive on the USB drive you chose:

You should cp this file to your working data folder and unzip. Paired end reads are in separate files; \*s\_n\_1\* and \*s\_n\_2\*):

-rw-rw-r-- 1 sbink users 4262974328 Nov 1 2010 BY4742\_MNase\_s\_1\_1.fastq

-rw-rw-r-- 1 sbink users 4262974328 Nov 1 2010 BY4742\_MNase\_s\_1\_2.fastq

Within most bioinformatics pipelines, you would check sequence quality using an app called FastQC at the start of analysis. However, I would suggest that you don’t. MNase-seq reads, in particular, FAIL FastQC spectacularly according to several metrics. In fact, ALL these failures – are a sign that the method has worked!: MNase-digested DNA *will* have a 5’ end A/T bias; nucleosome and TF protected DNAs are short relative to the Illumina input recommendations and *will* have significant 3’ end adapter contamination; nucleosome and TF protected DNAs *will* have odd k-mer bias and GC skews because of the sequence specificity of the proteins they were bound by; nuclease protected DNA fragments *will* occur multiple times and pretend to be duplicates.

***C. ALIGNING THE READS TO THE GENOME.***

We will use Bowtie1 (<http://bowtie-bio.sourceforge.net/manual.shtml>) Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10:R25.

Note: do not use Bowtie2 here! Bowtie1 is specifically written to effectively handle very short NGS reads.

**C1. Indexing.**

Bowtie needs to use a pre-built INDEX of a reference genome. I have already prepared an index of the “SacCer3” reference genome – which was used by Rossi et al., (2021). SacCer3 was published in 2011 as release R64\_1\_1. I from the NCBI. The index consists of six .ebwt files (“bw” refers to the Burroughs-Wheeler alignment algorithm used by Bowtie)

S\_cer\_full\_refseq.1.ebwt

S\_cer\_full\_refseq.2.ebwt

S\_cer\_full\_refseq.3.ebwt

S\_cer\_full\_refseq.4.ebwt

S\_cer\_full\_refseq.rev.1.ebwt

S\_cer\_full\_refseq.rev.2.ebwt

If you would like to find out what’s in the index, type within the relevant directory:

>bowtie-inspect --summary indexes/S\_cer\_full\_refseq

This should list all the chromosome (and plasmid) sequences present in the index.

**C2. Running the Bowtie aligner**

The relevant pair of .fastq files containing your paired-end-reads for a particular sample should be in the same directory as the index files, issue the command to load Bowtie (if you haven’t already):

Running Bowtie requires a command in the form:

>bowtie [options] [read 1 input file] [read 2 input file] [output file].

Here is the recommended command for the MNase reads:

>bowtie -n 0 --trim3 11 --maxins 1000 --fr -k 1 -p 4 --no-unal --sam -x S\_cer\_full\_refseq -1 BY4742\_MNase\_s\_1\_1.fastq -2 BY4742\_MNase\_s\_1\_2.fastq BY4742\_MNase\_unsorted.sam

The command above breaks down as:

bowtie “Run Bowtie in this directory...with the following options”

-n 0 “Find alignments with no mismatches”

--trim3 11 “Trim 11bp off the 3’ end of a 36bp read to leave 25bp”

--maxins 1000 “Throw away spuriously large apparent chromatin particles”

--fr “Show me read pairs with sensible F and R strand pairing - i.e. read-pairs that look like chromatin particles”

-k 1 “Output just one (the BEST) alignment for the read pair”

-p 4 “Use 4 processor cores/threads (set to 256 to make Ian M cry)”

--no-unal “Don’t output any unaligned reads”

--sam “Output the alignment list in .sam format”

-x “Use this genome index”

-1 “Following file contains the first read sequences”

-2 “Following file contains the other end sequences”

Output.sam “Name for the output file”

The precise command you use depends on the type of data, organism, read length etc. There are only two really important factors here:

1. The reads are trimmed from the 3’ end to remove adaptors and overlaps in very small nuclease protected DNA species. Even with short GAIIx reads, we trim back to 25bp. You might have heard of an app called Trimmomatic which removes adaptors prior to processing NGS reads? There’s no point using it here; the same end-product is achieved by trimming in bowtie – any remaining adaptor contaminated reads simply fail to align to the genome and get binned anyway.

2. The –fr flag makes sure that the reads point towards each other as would be expected if they had derived from opposite ends of a nuclease resistant “chromatin particle”.

Once you hit return, bowtie will run silently for ~3 min and will eventually deliver your .sam file and blurt out something like this on the screen:

# reads processed 253435343

#reads with at least one reported alignment 18876778 (72.08%)....

Anything with this methodology above 60% is good; anything lower, you might want to check that you sequenced the right organism!

**C3. The output files - .sam**

Each .sam output file (Li *et al.,* 2009 *Bioinformatics*. 25: 2078–2079) contains a list of all the read pairs from your two .fastq input files which Bowtie managed to align to the genome plus a variety of information concerning the alignment. One line from a .sam file will look something like this:

D3P26HQ1:178:D1A3WACXX:1:1101:11793:2878 147 gi|330443391|ref|**NC\_001133**.9| **135529** 255 28M = 135397 **160** TGCAGGTGTATTGCTGAGGGAATTCGGA GGGGGGGGHE?GFIGHHHHFDDB;F@@@ XA:i:0 MD:Z:28 NM:i:0

The key data for Kent lab MNase/DNaseI-seq technology are THREE simple bits of information from this line (these items of information are underlined):

1. The chromosome the read pair aligned to. NC\_001133 (or yeast chrI) in the above.

2. The position (in bp) at which the forward strand read began to align to that chromosome. 135529 in the above.

3 The distance (in bp) between the pair of reads (ISIZE - insert size). 160 in the above.

Items 1+2 tell us WHERE the chromatin particle giving rise to this read pair mapped in the genome; Item 3 tells us how much DNA was protected from MNase/DNase cleavage within the chromatin particle *in vivo* - i.e. WHAT the particle was.

**C4. Sorting read-pair alignments into chromosome order: samtools.**

The current .sam files list alignments in random order (essentially in the order that clusters were detected by the Illumina sequencer). We need to sort these into a more logical order. samtools (again Li et al., 2009) is a suite of utilities for manipulating .sam files.

Then issue the command (based on the outfile from bowtie above):

>samtools sort BY4742\_MNase\_unsorted.sam -o BY4742\_MNase.sam

Once you hit return, samtools will run for ~5 min, occasionally telling you what it’s doing to Nick’s hard drives, and will eventually deliver your new .sam file. If you open the new file (use less) you’ll find all the reads are now in chromosome and position order.

***D. PLOTTING CHROMATIN PARTICLE FREQUENCY DISTRIBUTIONS.***

**D1. Sorting aligned read-pair data into “chromatin particle”/PartN classes and plotting frequency distribution across the genome.**

The sequence data in the sorted .sam files represents all the nuclease-resistant fragments with ISIZE ranging from about 50bp up to ~600bp. Remember, mono-nucleosomal DNA will have an ISIZE of ~150bp. We now chop the data into distinct ISIZE classes which we designate with a “PartN”; the “Part” stands for nuclease resistant chromatin particle, and “N” is a defined value of ISIZE; e.g., Part150 data would represent likely mono-nucleosomal sequences.

This processing step is performed by a Perl script called SAM2PartN\_genome\_sgr\_full\_v22\_x.plx.

Hopefully you created a unique folder for this script (if not do it now and mv the script into it). Now also mv your sorted .sam file into the same folder. The script is hard-coded (parameters below) to deliver certain PartN outputs, but you can edit the script anyway you want.

$bin\_width - The histogram/distribution bin width value (your genomic *x*-axis) in bp. Default is 10bp.

@partn - A single value or comma delimited list of "chromatin particle size"/ISIZE/PartN. Default is (50,…,450) in 25bp steps. This range (although slightly overlapping at higher values) lets you plot lovely smooth surface graphs later on!

$pwind - The "particle window" +/- % of $partn. Default is 0.2 (+/-20%).

Within the script directory type:

>perl SAM2PartN\_genome\_sgr\_full\_v22\_x.plx

The script should output lots of useful information on the screen (you might want to copy this into a README.txt when the script finishes) and will deposit a series of .sgr files into a time stamped /out directory. Each of these .sgr files contains the frequency distribution of paired-read mid-point positions at a defined PartN value for every chromosome ID, concatenated to describe the whole genome! The frequency distributions will also have been lightly smoothed to a 3-bin moving average (“3MA”). Why the mid-point position?! For Part150 data this is equivalent to a nucleosome **dyad** position in the nucleosome crystal structure. This is traditionally what all MNase-seq plots show, and we have just expanded this to all PartN classes. You will notice that the binning resolution and PartN value are explicitly shown in the output filenames.

**D2. What is an .sgr file?!**

The .sgr is an old Affymetrix format called a sequence graphic file. If you open one in a text editor you’ll see something like:

:

chr9 12135 4

chr9 12150 4

chr9 12165 7

chr9 12180 10

chr9 12195 12

chr9 12210 44

chr9 12225 101

chr9 12240 108

chr9 12255 41

chr9 12270 21

chr9 12285 19

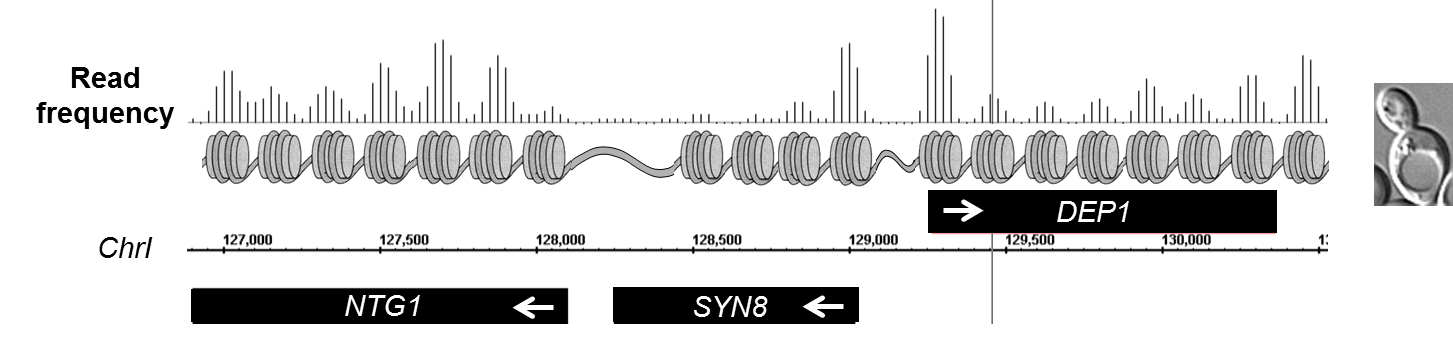
chr9 12300 8

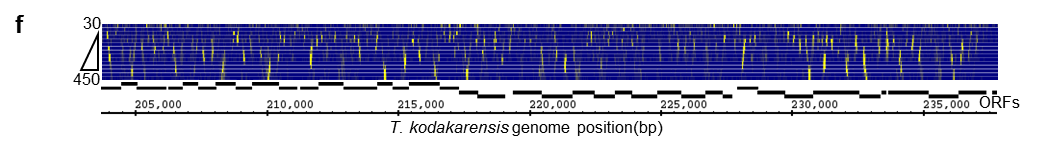
chr9 12315 9

chr9 12330 7

:

An .sgr file is just a three-column tab-delimited text file. Column 1 lists the chromosome ID; column 2 lists the bin position along that chromosome in bp; column 3 lists the frequency of sequence read mid-point positions found in that bin within the aligned data. Spot the peak/nucleosome!!! Ultimately the .sgr is just data values for plotting graphs that look like these in Genome Browser software:





Nick uses this file format because it is logical, human-readable and works with all his down-stream scripts. Other bioinformaticians either use .wig files which are hard for humans to read or .bedGraph files which are bulkier than .sgr. The .sgr format is accepted by the Integrated Genome Browser **(try plotting the Part50 and Part150 data**), but for plotting in the UCSC Genome Browser or IGV you should convert .sgrs to .bedGraph. This is a four-column tab-delimited text format which can be generated using the script sgr\_to\_bedgraph.plx.