


ATAC-seq

The analysis on this sheet are based on the results of the `nf-core/atacseq` pipeline (c.f. general task sheet). In section 1 you will run the pipeline. Section 2 utilizes the alignment files output by the pipeline to run a more comprehensive analysis using the `ChrAccR` R package. Section 3 involves more detailed analysis.




Reminder

- Make sure you work inside a screen
- Document the steps including core commands used and the most important output in you workshop notebook. Tasks that need reporting to your notebook are marked with  symbol.

1 Nextflow ATAC-seq pipeline

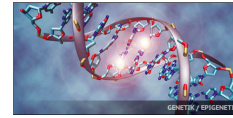
- ▲ Make sure you run this task on the master node
- ▲ We limited the Nextflow Java virtual machines memory into `NXF_OPTS='-Xms200m -Xmx300m'`. Do NOT change this.

1.1 Running the pipeline

1. **Organize your raw data** it is a good practice to have the raw data placed in one folder:
 - a) List the raw fastq files under `/vol/COMPEPIWS/data/reduced/ATAC-seq/`
 - b) How many files per cell_type per time point per replicate are there? What does the suffix `*_R{1,2}` mean? 
2. **Create a samplesheet** The pipeline uses a samplesheet as input in order to organize the pipeline workflow. This sheet should contain the input file locations and some basic sample annotation.
 - a) Enter your working directory `/vol/COMPEPIWS/groups/<group>/tasks`. Here, create a directory called `nextflow_run` and enter that directory.
 - b) From the pipeline documentation: find out what format the sample sheet should have. Prepare a file `samplesheet.csv` that contains the following information for the samples you have: `sample`, `fastq_1`, `fastq_2`, `replicate`. Make sure you include the correct path for your input files. 
3. **Run the ATAC-Seq pipeline** As you might noticed from the documentation, you can provide parameters on the command line or add them to a configuration file which you pass to `(-c)` option. We have prepared a config file for you, have a look to the parameters in `/vol/COMPEPIWS/pipelines/nextflow/assets/nf-core/atacseq/atacseq.config`. We will use singularity as a profile.
 - a) Report based on the config file: 
 - i. is the data paired end or single end?
 - ii. which steps of the pipelines are skipped?
 - iii. why is the blacklist file used? What do these regions represent?
 - b) Load Conda environment:

```
source /vol/COMPEPIWS/conda/miniconda3/bin/activate /vol/COMPEPIWS/conda/miniconda3/envs/atacseq
```
 - c) Run the pipeline (will take around 30 minutes to complete):
Note: please talk to your mentor to check the samplesheet before you start running the pipeline

```
nextflow run nf-core/atacseq -r 2.1.2 -profile singularity  
--input samplesheet.csv --outdir /PATHTO/OUTDIR -params-file  
/vol/COMPEPIWS/pipelines/param_files/atacseq_params.json
```



4. watch your jobs using `squeue`.
5. An output directory names `results` will appear in the directory that you used to run the analysis. Note the complete path to that directory.
6. inspect the pipeline report generated in `results/pipeline_info/execution_report.html`. How long did your pipeline run to completion? Which task used the most CPU resources? Which task used the most memory? Which task took the longest to complete?

1.2 Quality Control (QC)

1. Locate the directory of the `fastqc` reports. Inspect the reports
 - a) Which read length was used in sequencing?
 - b) What does the “Per base sequence quality” tell you? Are there any samples that fail this check? If so, why?
 - c) What does the “Per base sequence content” tell you? Are there any samples that fail this check? If so, why?
2. Locate the aligned read files (BAM files) (`results/bwa/mergedLibrary`). Note the directory. You will use it again later.
 - a) run `samtools flagstats` for each BAM file and report how many reads were aligned.
3. Locate the peak calling results (`results bwa/merged_library/macs2/narrow_peak/`).
 - a) How many peaks were called for each sample?
 - b) Explain the difference between the `*_peaks` and the `*_summits` files.
4. Locate the MultiQC report (`/results/multiqc/narrowPeak/multiqc_report.html`) and inspect it
 - a) Which sample has the highest read duplication rate? *Hint*: section: “LIB: FastQC (raw)”
 - b) Which sample has the most peaks called? *Hint*: section: “MERGED LIB: MACS2 peak count”
 - c) Which sample has the lowest FRiP score? *Hint*: section: “MERGED LIB: MACS2 peak FRiP score”
 - d) Specify the percentage range of peaks overlapping with promoters. *Hint*: section: “MERGED LIB: HOMER peak annotation”
 - e) What is the difference between the “MERGED LIB” and the “MERGED REP” sections?

2 Integrative analysis using ChrAccR

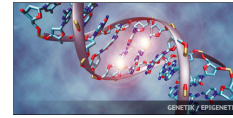
Make sure you run this task on a worker node

We will use the comprehensive ChrAccR R package to analyze the dataset. You can find documentation on the package at <https://greenleaflab.github.io/ChrAccR/>. To familiarize yourself with the basic workflow, we recommend reading the “Overview” vignette.

2.1 Running the pipeline

1. Prepare a sample annotation sheet with details on all samples that contains the following columns:
 - `sampleId`: a unique sample identifier
 - `tissue`: liver or kidney
 - `time`: timepoint at which the sample was taken
 - `replicate`: replicate 1 or 2
 - `bamFile`: name of the bam file


You should be able to derive all this information from the bam file names (Nextflow pipeline output)



2. Configure the analysis. Make sure to set the appropriate analysis parameters for the following:

- enable differential analysis for the `tissue` column


Hint: Take a look at `?getConfigElement`. Useful options include `differentialColumns`

3. Specify a list of objects containing `GenomicRanges` objects with region sets of interest :

- a) We configured a base list of regions for you. You can find the R object at `/vol/COMPEPIWS/data/annotation/regionSetList.rds` (*Hint:* `readRDS(...)`). Which region sets does this list contain? Why are these region sets interesting for chromatin accessibility?
- b) Locate the set of consensus peaks output by the Nextflow pipeline (`results/bwa/merged_library/macs/narrowPeak/consensus/consensus_peaks.mLb.cIN.bed`). This file contains a consensus set of peaks determined by merging peaks from all samples. Load this file into a `GenomicRanges` object and add it to the region set list under the key `'consensusPeaks'`.

4. Run the analysis using `run_atac(...)`. Make sure that you specify the region set list you constructed in the previous step. This analysis takes about 20 minutes.

2.2 Interpreting the output

If the pipeline completed successfully, it will create a `reports` directory in its output directory. Open the main report located under `reports/index.html` and browse through the results. 

1. Summary

- a) How many regions were annotated with chromatin accessibility information for each region set?
- b) Do you see samples with low numbers of fragments?
- c) Why is there a second “hump” downstream of the TSS in the TSS profile plots
- d) Are there any samples with a bad TSS enrichment? Which sample has the lowest TSS enrichment score?

2. Filtering

- a) How many regions of each type were removed from the dataset? Why were they removed?

3. Normalization

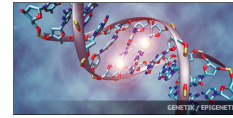
- a) What type of normalization was applied?
- b) What effect did the normalization have on samples with lower fragment numbers that you identified in the Summary (task 1b)?

4. Exploratory analysis

- a) Inspect the dimension reduction and clustering plots. Which region set best discriminates between liver and kidney samples? Explain your choice.
- b) Can the samples be separated based on developmental time?
- c) Which TF motifs exhibit the largest variance across all samples in the dataset?

5. Differential analysis








- a) Inspect the differential peaks between kidney and liver. Peaks in which tissue are generally more accessible?
- b) Name two TF motifs that are globally more accessible and two motifs that are less accessible in kidney compared to liver.

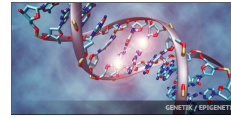




3 Exploratory and differential analysis

▲ Make sure you run this task on a worker node

The ChrAccR analysis generated a DsATAC object that contains useful data for downstream analysis, including sample annotation, accessibility counts summarized over regions of interest, fragment and insertion sites, and more.

1. Load the filtered dataset (located in your ChrAccR output directory under `data/dsATAC_filtered`) from the analysis using `loadDsAcc(...)`. Inspect the dataset :
 - a) How many samples does it contain?
 - b) How many fragments does each sample have?
 - c) Which region sets were summarized?
2. Plot a heatmap showing normalized counts per sample and peak for the 100 most variable peaks across the dataset. Use `log2 CPM` normalized counts for displaying.  *Hint: `edgeR::cpm` or `ChrAccR::transformCounts` for CPM normalization, `matrixStats::rowVars` for finding the most variable peaks and `pheatmap::pheatmap` for plotting.*
3. Export the peak and promoter count matrices as tab-separated table for later integrative analysis *Hint: `getCounts(..., write.table(...)`*
4. Compute chromVAR motif activities for the consensus peak set using the JASPAR vertebrate motif annotation. *Hint: `getChromVarDev(..., motifs="jaspar_vert")`.*
5. Plot the variability of accessibility in motifs  *Hint: `computeVariability` and `plotVariability` from the chromVAR package. Which are the 5 most variable motifs?*
6. Plot a heatmap showing the deviation Z scores of the 20 most variable motifs across all samples in the dataset. *Hint: `deviationScores` from the chromVAR package and `pheatmap` from the pheatmap package.*
7. Identify differentially accessible peaks between kidney and liver samples
 - a) compute a table summarizing the differential statistics for each peak. *Hint: `getDiffAcc(...)`*
 - b) Add a column to the table that classifies each peak as either: (i) not differential, (ii) more accessible in kidney, (iii) more accessible in liver. Use the following thresholds for this categorization: (i) adjusted p-value less than 0.01, (ii) log fold-change greater than 3. How many regions of each type do you have? 
 - c) Plot an volcano plot highlighting the differential peaks.  *Hint: `ggplot`, `geom_point`*
 - d) Add genome coordinates to the differential annotation table. *Hint: `getCoord(...)`*
 - e) For each peak, annotate the name of gene whose TSS is closest to that peak. Also annotate the distance to that gene. *Hint: `ChrAccRAnnotationMm10::getGeneAnnotation(anno="gencode_coding", type="tssGr")` returns a GRanges object of TSS coordinates. `distanceToNearest(...)` computes the closest distances, given two GRanges objects.*
 - f) Export the table of differential peaks annotated with the respective genes as tab-separated value file
 - g) Create a violin plot showing the distribution of distances to the nearest TSS for each category. Limit the plot to a maximum of 100kb distance.  *Hint: `ggplot(...)` + `geom_violin()`*
 - h) (optional) Are differential peaks further from genes than non-differential peaks? To answer this, compute whether TSS-distances are significantly greater for the kidney and liver-specific peaks compared to the non-differential peaks. Compute p-values using `wilcox.test(...)`. 
 - i) Export the sets of differential peaks as separate BED files for peaks more accessible in kidney and peaks more accessible in liver
8. Combine samples corresponding to liver and kidney tissue, respectively:
 - a) Create a summary dataset that combines all samples of a tissue into one aggregate sample. *Hint: `mergeSamples(...)`*



- b) For finer resolution, create a `GenomicRanges` object containing 200bp-tiling windows. Add this region set to the tissue dataset created in the previous step. *Hint: `regionAggregation(..., signal="insertions")`*
 - c) Export count tracks that can be viewed in the IGV genome browser using the 200bp-tiling regions from the previous step. *Hint: `exportCountTracks(...)`*
 9. From the tissue-combined dataset, plot aggregate TF footprints for the 4 motifs you identified in task 5b (section 2.2)  *Hint: `getMotifFootprints(...)`*
 10. Explore the data in IGV
 - a) Open an IGV browser window using the `mm10` genome assembly
 - b) load the following tracks:
 - The consensus peak set (BED file)
 - The aggregated kidney and liver tracks you created in task 8c. Use the autoscale option to normalize these tracks
 - The differential peak sets (BED files) created in task 7i
 - c) Save this IGV session to use again later.
 - d) Explore the regions around the most differential peaks (according to the ranking statistic from the differential table created in task 7). What do you observe? Do you see more changes at promoters or distal peaks? 

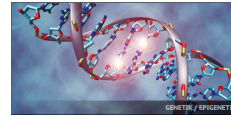
4 Prepare data for downstream integrative analysis

In order to facilitate integrative data analysis and sharing of your results with other groups, please deposit your main output files using the following instructions:

1. Create a folder for your group in `/vol/COMPEPIWS/groups/shared/ATAC-seq`.
2. Copy (**not** link) and organize the following files:
 - consensus peak set (BED file)
 - The count matrices created in task 3
 - Aggregated kidney and liver tracks you created in task 8c
 - The differential peak set table (TSV file) created in task 7f
 - The differential peak sets (BED files) created in task 7i

Here's how the directory structure should look like:

```
/vol/COMPEPIWS/groups/shared/ATAC-seq
|
|__ atacseq1
|   |__ peaks
|   |__ signal
|   |__ counts
|   |__ differential
|
|__ atacseq2
|   |__ peaks
|   |__ signal
|   |__ counts
|   |__ differential
|
```



```
|__ atacseq3
    |__ peaks
    |__ signal
    |__ counts
    |__ differential
```

3. Prepare one slide explaining how you organized the files, the content, and how they can be used.
4. Make sure that only people of your group can modify the file, but all can read it. *Hint: chmod*