



Maris Šiljak, BSc

Plasma DNA Cancer Predictive Analytics

Master's Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Denis Helic

Institute of Interactive Systems and Data Science (ISDS)

Co-supervisor

Univ.-Prof. Dr.med. Michael Speicher

Institute of Human Genetics

Medical University of Graz

Graz, October 2021

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Abstract

This thesis investigates the potential to distinguish between healthy and cancer patients moreover between different cohorts of cancer patients by combining medical and computer science techniques. The distinction between patients was performed by utilising the machine learning classification method. As the prerequisite for the classification, we processed the patient data, generated specific genomic location datasets and finally extracted and normalised the signals describing the patient data. The evaluation was performed based on the classification results and extracted signals behaviour, and it showed great potential for the distinction of patients. We conclude that the investigated genomic locations are directly related to cancer biology. The classification potential depends directly on the quality of the genomic location dataset and the patient tumour fraction.

Contents

Abstract	iii
1 Introduction	1
1.1 Processing of Biosamples	2
1.2 Generation of DHS Datasets	3
1.3 Extraction and Normalisation of Biosample Coverage Signals at DHSs	6
1.4 Machine Learning Classification	7
2 Related Work	9
2.1 Liquid Biopsy	9
2.2 DNA Sequencing and Sequencing Technologies	10
2.2.1 NGS	12
2.3 DNase-seq	12
2.4 Cell-free DNA	14
2.5 File Formats	15
2.5.1 FASTQ	15
2.5.2 SAM/BAM	16
2.5.3 BED	17
2.6 Biosample Datasets	17
2.6.1 Colorectal Cancer (CRC)	18
2.6.2 Prostate Cancer (PC)	18
2.6.3 Healthy Controls	18
3 Methods	20
3.1 Processing of Biosamples	20
3.1.1 Conversion	22
3.1.2 Merge Lanes	22
3.1.3 Adapter Trimming	22

Contents

3.1.4	Quality Report	24
3.2	Clustering of DNase-seq Data	25
3.2.1	Retrieval of Vierstra DNase-seq Data for Clustering . .	26
3.2.2	DNase-seq Data Custom Clustering Method	28
3.2.3	DNase-seq Data Bedtools-based Clustering Method . .	31
3.3	Processing of Pre-clustered DHS Datasets	33
3.3.1	Collecting and Lifting-over Sheffield Duke Clusters . .	33
3.3.2	Statistical Preprocessing of Meuleman Intermediate DHS Matrix	39
3.3.3	Splitting Meuleman DHS Components	47
3.4	Coverage Signal Normalisation Methods	48
3.4.1	Surrounding Regions Median	49
3.5	Multi-class Classification	51
3.5.1	Feature Extraction	51
3.5.2	Model Training Method	52
4	Results and Discussion	53
4.1	Coverage Signal Normalisation Method	53
4.2	Coverage Plot Structure	55
4.3	Evaluation of DHS Datasets	58
4.3.1	Evaluation of Hematopoietic-based DHS Datasets . . .	58
4.3.2	Evaluation of Prostate and Prostate Cancer-based DHS Datasets	61
4.3.3	Evaluation of Digestive-based DHS Datasets	65
4.4	Impact of Biosample Tumor Fraction (TF) on Coverage Signal	73
5	Conclusion	75
5.1	Future Work	76
	Bibliography	77

List of Figures

1.1	Representation of DNase-seq processed biosamples (rows) with detected DHSs (top). Three compiled clusters exemplify three representative DHSs (middle). Cluster (representative DHS) exemplifying calculated metrics (start, end, core_start and core_end) (bottom) [Meuleman et al., 2020]	5
1.2	Coverage signal (2k window) of one biosample for a DHS dataset. Position 0 (x-axis) reflects the midpoint or peak of respective DHSs. Coverage (y-axis) represents the mean coverage of respective DHSs	7
1.3	General structure of time series data ¹	8
2.1	Liquid biopsy procedure and clinical application [Pinzani et al., 2021]	10
2.2	Evolution of sequencers over time ²	11
2.3	DNase-seq workflow. Detecting DHSs and building DHS genome-wide peak model for a biosample. Adapted from [Ling and Waxman, 2013]	13
2.4	Blood plasma stream and constituting parts with the focus on cfDNA and ctDNA [Hahn et al., 2019]	14
2.5	FASTQ file structure legend ³	15
2.6	Phred-33 mapping quality legend ⁴	16
2.7	Phred-33 quality scores ⁴	16
3.1	uBAM2FASTQ pipeline workflow	21
3.2	Fragmentation of a chromosome on fragments (molecules). Adapter ligation (attaching) to both ends (5' and 3') of fragments. Adapted from [Bioinformatics, 2016]	23

List of Figures

3.3	Read length shorter than the fragment length (top). Read length longer than the fragment length (bottom). Adapted from [Bioinformatics, 2016]	23
3.4	Interface of <i>Human Body Map</i> . Colon and large intestine organs selected ⁵	26
3.5	<i>Human Body Map</i> displaying available sequencing data for selected colon and large intestine ⁶ . DNase-seq column is highlighted	27
3.6	Example of DNase-seq output file structure	28
3.7	Merge and sort of DNase-seq files according to their start and end position	29
3.8	2D representation of DHSs on a portion of genome. Red lines express brake condition for <i>Custom Clustering</i> method	30
3.9	2D demonstration how the four cluster metrics (<i>start, end, core_start, core_end</i>) are calculated	31
3.10	2D demonstration how <i>bedtools multiinter</i> calculates overlaps ⁷	32
3.11	Example of <i>bedtools multiinter</i> output format	32
3.12	Interface for Duke DHS Cluster Database ⁸ depicting tissue type as column name and belonging cell lines as its rows . . .	34
3.13	Liver cluster 1066 ⁹ with an adequate tissue-specific accessibility pattern (blue bars in comparison to others)	35
3.14	Liver cluster 1115 ¹⁰ with an adequate tissue-specific accessibility pattern (blue bars in comparison to others)	35
3.15	Boxplot of Meuleman DHS matrix biosamples (20 - 40) and their raw DHS accessibility values	40
3.16	KDE of Meuleman DHS matrix biosamples (20 - 40) and their raw DHS accessibility values	41
3.17	Boxplot of Meuleman DHS matrix biosamples (20 - 40) and their winsorized DHS accessibility values by 5%	42
3.18	KDE of Meuleman DHS matrix biosamples (20 - 40) and their winsorized DHS accessibility values by 5%	43
3.19	Boxplot of Meuleman DHS matrix biosamples (20 - 40) DHS accessibility values. Winsorization (5%) and quantile normalisation of accessibility value distributions to each other applied	44

List of Figures

3.20	KDE of Meuleman DHS matrix biosamples (20 - 40) DHS accessibility values. Winsorization (5%) and quantile normalisation of accessibility value distributions to each other applied	45
3.21	Boxplot of Meuleman DHS matrix biosamples (20 - 40) DHS accessibility values. Winsorization (5%), quantile normalisation and MinMax scaling between 0 and 1 of accessibility values applied	46
3.22	KDE of Meuleman DHS matrix biosamples (20 - 40) DHS accessibility values. Winsorization (5%), quantile normalisation and MinMax scaling between 0 and 1 of accessibility values applied	46
3.23	Meuleman DHS Components file structure ¹¹	48
3.24	Surrounding regions schema. Black curve represents coverage signal of one biosample which is normalised by median value of two <i>surrounding regions</i> (red)	49
3.25	DHS distance to neighbouring peak in 20k window	50
4.1	Coverage plot of two biosamples (<i>NPH_001</i> and <i>C123_4</i>) with raw coverage values for the DHS dataset <i>hematopoietic_270</i> . .	54
4.2	Coverage plot of two biosamples (<i>NPH_001</i> and <i>C123_4</i>) with SRM normalised coverage values for the DHS dataset <i>hematopoietic_270</i>	55
4.3	Two coverage plot examples. Coverage plot of 10k ROI window (top plot). Coverage plot of 2k ROI window with three labeled parts (bottom plot).	57
4.4	Coverage plot of Duke_hematopoietic_200 DHS dataset in 2k window	59
4.5	Duke_hematopoietic_200 model classification report and confusion matrix	59
4.6	Coverage plot of Duke_hematopoietic_270 DHS dataset in 2k window	60
4.7	Duke_hematopoietic_270 model classification report and confusion matrix	61
4.8	Coverage plot of Duke_hematopoietic_200 DHS dataset in 2k window	62

List of Figures

4.9	Duke_prostate_664 model classification report and confusion matrix	62
4.10	Coverage plot of Duke_prostate_cancer_1135 DHS dataset in 2k window	63
4.11	Duke_prostate_cancer_1135 model classification report and confusion matrix	63
4.12	Coverage plot of Duke_prostate_cancer_LNCaP_andro_15236 in 2k window	64
4.13	Duke_prostate_cancer_LNCaP_andro_15236 model classification report and confusion matrix	65
4.14	Coverage plot of Duke_liver_437 DHS dataset in 2k window	66
4.15	Duke_liver_437 model classification report and confusion matrix	66
4.16	Coverage plot of clustering_bedtools_Vierstra_digestive_small_70p_62835 DHS dataset in 10k window	67
4.17	Coverage plot of clustering_bedtools_Vierstra_digestive_small_70p_62835 DHS dataset in 10k window without PCs	68
4.18	clustering_bedtools_Vierstra_digestive_small_70p_62835 model classification report and confusion matrix	68
4.19	Coverage plot of clustering_custom_Vierstra_digestive_small_70p_52420 DHS dataset in 10k window	69
4.20	Coverage plot of clustering_custom_Vierstra_digestive_small_70p_52420 DHS dataset in 10k window without PCs	70
4.21	clustering_custom_Vierstra_digestive_small_70p_52420 model classification report and confusion matrix	70
4.22	Coverage plot of prep_Meuleman_colon_win_02_5p_05_top_1k DHS dataset in 2k window	71
4.23	Coverage plot of prep_Meuleman_colon_win_02_5p_05_top_1k DHS dataset in 10k window	72
4.24	prep_Meuleman_colon_win_02_5p_05_top_1k model classification report and confusion matrix	72
4.25	Coverage plot of Duke_hematopoietic_270 DHS dataset in 2k window with overlapping CRCs and their tumor fraction labels	73
4.26	Misclassification bar chart of Duke_hematopoietic_270 model	74

1 Introduction

Human genome is a large and complex structure. Its substantiality differs from human to human, but certain genomic regions are related to the health of a given person. We aim to exploit this fact and compose as specific datasets of genomic regions as possible for different cancer types. Specificity reflects how good a set of genomic regions describes a particular cancer type. This thesis focuses on a specific group of genomic regions assumed to be related to cancer biology, namely, DNase I hypersensitive sites (DHSs) [D'Antonio et al., 2017]. However, the role of DHSs in machine learning has not been widely reported. Therefore, we build our research topic based on DHSs and machine learning methods, with an ultimate goal to distinguish between healthy and cancer patients moreover between different cohorts of cancer patients (colon and prostate). In order to reach the ultimate goal, we need to reach **three subgoals**.

First subgoal relates to converting, merging, and trimming sequenced patient DNA data (biosamples). Modern DNA sequencers output files in a non-standard file format for our project. Therefore, we convert output files in an appropriate format. Modern DNA sequencers also perform sequencing in parallel producing multiple output files. Therefore, we merge output files. These output files contain adapter sequences (short DNA sequences attached to the actual DNA fragments) relevant to sequencers but not our analyses. They could even harm our future analyses. Therefore, we trim merged output files.

Second subgoal relates to the generation of DHS datasets. Currently, there is a lack of highly cancer type-specific DHS datasets. Therefore, we generate DHS datasets based on two types of data sources:

- DNase-seq Data - requires clustering in order to establish a dataset of representative DHSs

1 Introduction

- Pre-clustered DHSs - require domain expert validation and statistical preprocessing

Third subgoal relates to extraction and normalisation of biosample coverage signals at DHSs. Coverage stands for the number of times each base was sequenced. Coverage signal represents coverage values in a defined signal range (window). This subgoal links the previous two subgoals, such that we extract the coverage signals from biosamples processed in the **first subgoal** at DHSs generated in the **second subgoal**. However, as humans exhibit biological differences, comparing unnormalised biosample coverage signals results in incorrect data. Therefore, we introduce and perform a normalisation method to overcome the issue.

1.1 Processing of Biosamples

We have a collection of biosamples, which have been occasionally sequenced over a couple of years. Sequencing technology has changed with time, and so have the output files. Moreover, changes in the sequencing technology reflect deviations in the collected biosamples (e.g. parallel sequencing on four lanes instead of two, different adapter sequences to trim, output file format). Therefore, to handle the different scenarios and automate the uniform processing workflow, we develop a biosample processing pipeline called the *uBAM2FASTQ* pipeline. To speed up the processing of biosamples, we build this pipeline with multi-core logic supporting SLURM-based¹ clusters.

We incorporate the following external tools in the pipeline:

- *samtools* (version 1.13)
- *cutadapt* (version 3.4)
- *cat*
- *FastQC* (version 0.11.9)

Workflow of the pipeline consists of five steps:

¹<https://slurm.schedmd.com/documentation.html>

1 Introduction

1. **Conversion** - We define *FASTQ* as the standard file format for our biosamples. *FASTQ* is a text-based (human-readable) format for storing biological sequences (sequences of four DNA bases) (Subsection 2.5.1). However, most of our collected biosamples are stored in *uBAM* (unaligned BAM) format. *uBAM* is a binary-based format for storing biological sequences (Subsection 2.5.2). Therefore, we perform conversion of *uBAM* biosamples to project defined standard format, the *FASTQ*. We utilise the *samtools* suite to perform the conversion.
2. **Merge Lanes** - Modern DNA sequencers use multiple lanes to perform parallel sequencing producing multiple output files of one biosample. Therefore, we merge produced output files assembling the complete biosample file. We utilise the Unix default *cat* tool to merge lanes, and additionally, we implement multi-core logic supporting SLURM-based clusters.
3. **Quality Report** - We need to examine the quality of so far processed biosample files and provide insight into the completeness of the previous two steps. Therefore, we utilise *FastQC* to produce the quality report. This report contains statistics about the corruption of a biosample file or parts of it.
4. **Trim Adapters** - Adapters are attached to the actual DNA fragments during library preparation in the lab. They enable sequencers to detect DNA fragments but are of no use to us. Removing adapters is generally considered to be a good practice. Therefore, we utilise *cutadapt* to trim adapters.
5. **Quality Report** - We need to make sure the adapters are gone. Therefore, we utilise once again *FastQC* to produce the quality report. This report also contains statistics about adapter contamination.

1.2 Generation of DHS Datasets

Examining biosamples on the genome basis is practically impossible. Therefore, we narrow down the analysis to specific genomic regions assumed to be related to cancer biology, the DHSs. Unfortunately, we currently lack DHS datasets with high cancer type specificity. Therefore, we generate DHS datasets based on two types of data sources:

1 Introduction

- DNase-seq Data
- Pre-clustered DHSs

DNase-seq Data

DNase-seq is a specific DNA sequencing method used to detect DHSs in a biosample. The advantage of DNase-seq data compared to the other data source (pre-clustered DHSs) is its wide distribution and accessibility. For example, certain research groups perform DNase-seq on their biosample cohorts and distribute the data for further analyses [Vierstra et al., 2020]. However, the drawback of DNase-seq data is that we have to perform clustering ourselves to create representative DHS datasets.

For every DNase-seq processed biosample, one output file gets produced. These output files contain detected DHSs (rows) with their respective genomic chromosome, start and end positions (columns). Analysing the genome-wide distribution of contained DHSs genomic position displacements occur between different output files (different biosamples). DHSs are shifted to the left or the right compared to other biosamples, meaning no perfect overlap is visible. Displacements happen mainly due to biological nature.

1 Introduction

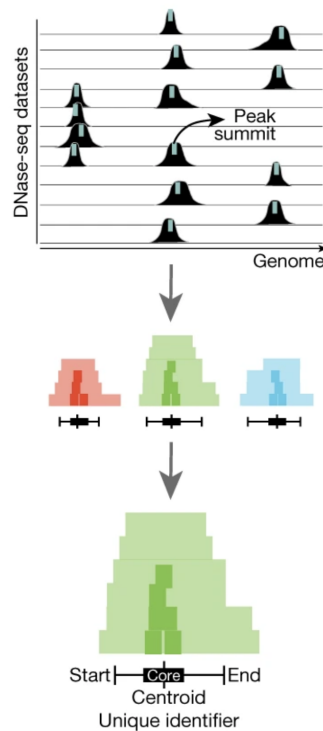


Figure 1.1: Representation of DNase-seq processed biosamples (rows) with detected DHSs (top). Three compiled clusters exemplify three representative DHSs (middle). Cluster (representative DHS) exemplifying calculated metrics (start, end, core_start and core_end) (bottom) [Meuleman et al., 2020]

To solve occurring displacements and create representative DHS datasets from DNase-seq data, we come up with two methods:

- Custom Clustering
- Bedtools-based Clustering

Pre-clustered DHSs

Pre-clustered DHSs are already established representative DHSs. Certain research groups perform DNase-seq on their biosample cohorts and go a step further and cluster the DNase-seq data with their clustering methods

[Meuleman et al., 2020, Sheffield et al., 2013]. Although both of these publications provide pre-clustered DHSs, the data slightly differs.

Data provided in **Sheffield et al., 2013** comes in tissue/cell line-specific format. Therefore, we pick specific tissues/cell lines targeting cancer types of our biosample cohorts.

Data provided in **Meuleman et al., 2020** comes in both tissue-specific and biosample-specific formats (as two separate files). Although they depict the same representative DHS dataset, we must preprocess them differently. Regarding tissue-specific DHSs, we split the DHSs into different files based on assigned tissue. Regarding biosample-specific DHSs, we perform statistical preprocessing on the DHSs, filtering out irrelevant DHS and targeting cancer types of our biosample cohorts.

1.3 Extraction and Normalisation of Biosample Coverage Signals at DHSs

Coverage signal extraction is the link between processed biosamples and generated DHS datasets. Coverage stands for the number of times each base was sequenced. Coverage signal represents coverage values in a defined signal window, whereas position 0 stands for DHS peak or midpoint. We have DHS datasets targeting specific biosample cohorts (HC, CRC and PC). Therefore, we extract coverage signals at all DHSs contained in a DHS dataset for all biosamples. Then, we average all extracted DHS coverage signals of a particular DHS dataset belonging to a particular biosample delivering one coverage signal per biosample.

1 Introduction

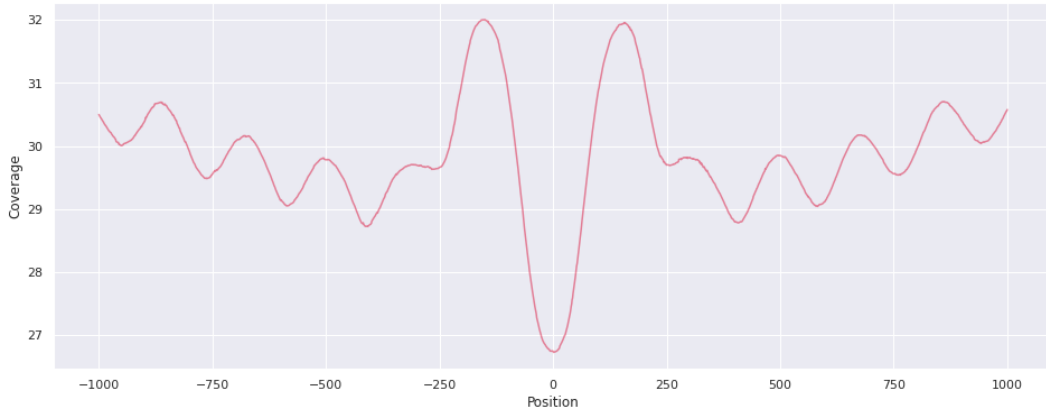


Figure 1.2: Coverage signal (2k window) of one biosample for a DHS dataset. Position 0 (x-axis) reflects the midpoint or peak of respective DHSs. Coverage (y-axis) represents the mean coverage of respective DHSs

To extract coverage, we utilise a Python package called *Pysam*². *Pysam* contains a ready-made coverage calculation function but no multi-core support. Therefore, we incorporate the package in our script and implement multi-core logic supporting SLURM-based clusters to speed up the coverage extraction.

Deviations in terms of mean coverage between biosamples are natural and expected things. The comparison between their unnormalised coverage signals results in incorrect data. Therefore, we introduce and apply a normalisation method called *surrounding regions median (SRM)*.

1.4 Machine Learning Classification

Now that we reached all three subgoals, we recall the ultimate goal of machine learning classification. Prior to the classification task, we require features of extracted coverage signals. For the feature extraction, we utilise a Python package called *tsfresh* (version 0.18.0). This package enables the automatic calculation of a large number of time series features, and it also

²<https://pysam.readthedocs.io/en/latest/index.html>

1 Introduction

contains statistical tests to filter out features with low importance and explanatory power.

The main requirement for the package is to work with time series data. If we recall the general time series structure, it is clear that our coverage signals are deducible to time series.

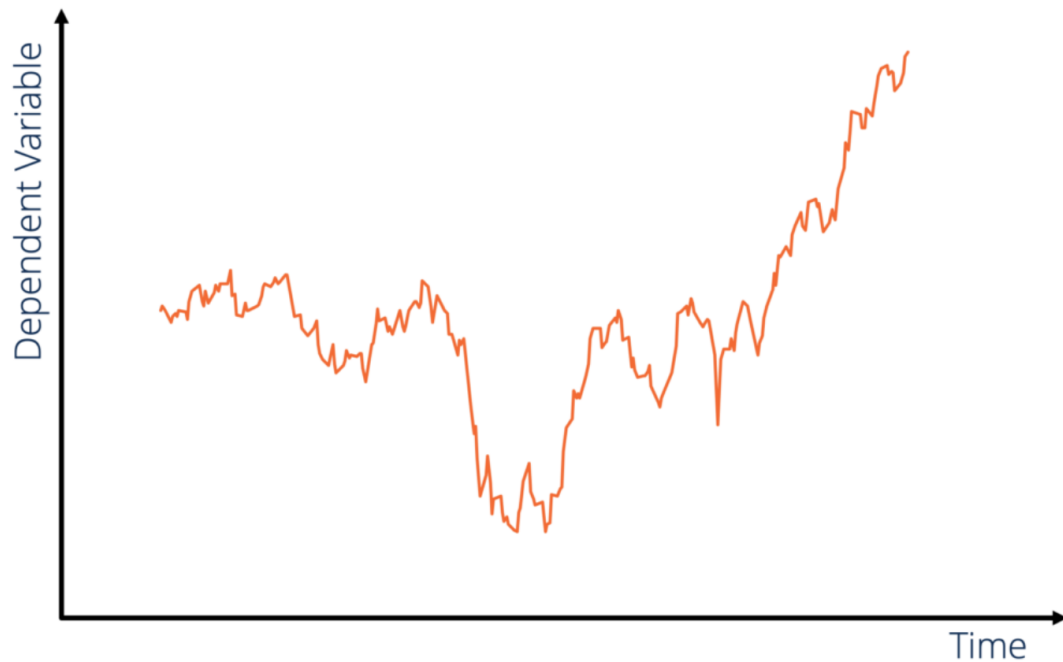


Figure 1.3: General structure of time series data³

Therefore, we interpret *coverage* (y-axis on coverage plots) as the *dependent variable*. Likewise, we interpret *position* (x-axis on coverage plots), captured in constant intervals (one base), as the *time*.

We utilise an ensemble learning method called *Random Forests* [Breiman, 2001] for the classification task. It is our machine learning method of choice, as it delivers a relatively good accuracy for a reasonable amount of time.

³<https://corporatefinanceinstitute.com/resources/knowledge/finance/time-series-data-analysis/>

2 Related Work

2.1 Liquid Biopsy

The blood plasma of humans contains small fragments of DNA referred to as circulating cell-free DNA (cfDNA) (Section 2.4). The hematopoietic system, particularly the white blood cells, is the predominant cfDNA contributor in healthy individuals. However, under certain physiological or pathological conditions, other organs may substantially contribute to cfDNA, which can be leveraged for research applications and noninvasive diagnostic purposes [Heitzer and Speicher, 2018]. Because of its diagnostic potential, the analysis of plasma DNA is frequently referred to as liquid biopsy. Hence, we require only the blood sample from a patient in contrast to invasive methods.

2 Related Work

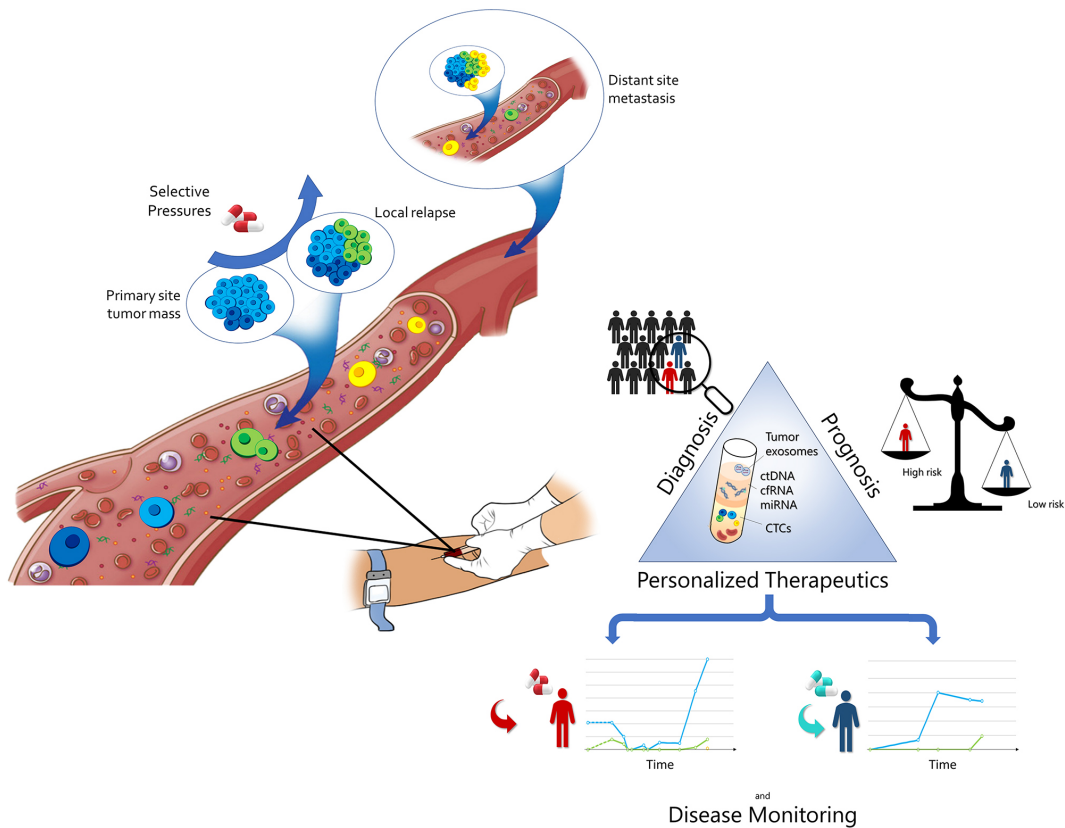


Figure 2.1: Liquid biopsy procedure and clinical application [Pinzani et al., 2021]

2.2 DNA Sequencing and Sequencing Technologies

DNA sequencing is the process of determining the nucleic acid sequence (sequence of nucleotides in DNA) [Pareek, Smoczynski, and Tretyn, 2011, Mitchelson, 2005]. Devices that enable this process are called *DNA sequencers*. Over time, different generations of DNA sequencers have been developed for different purposes [Pradhan et al., 2019]:

- First Generation
- Second Generation (Next Generation)

2 Related Work

- Third Generation
- Fourth Generation

The development timeline of three generations of sequencers:

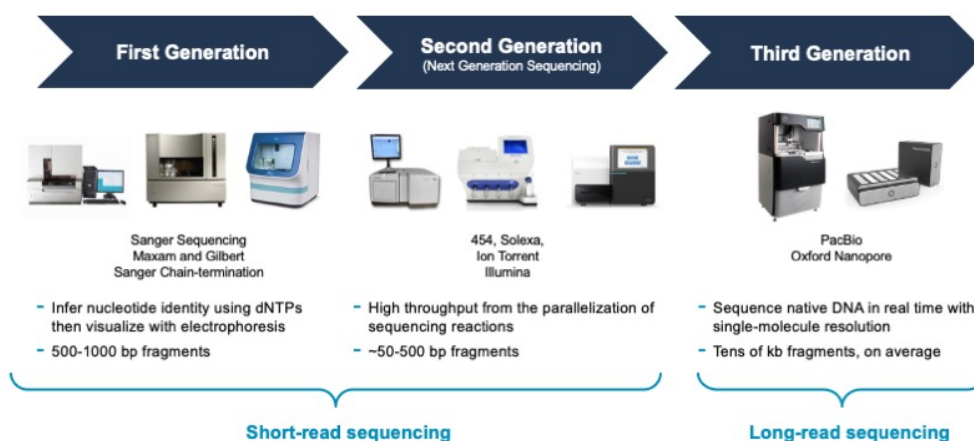


Figure 2.2: Evolution of sequencers over time¹

Due to the biological nature of cfDNA (short fragments), the fourth generation used for long-read sequencing is of no importance to us. Second generation or next-generation sequencing (NGS) devices are of particular interest for us, offering relatively good cost/speed balance. All of our biosamples were sequenced on NGS devices.

Two general types of sequencing exist:

- Whole-genome sequencing (WGS) - utilised to capture information on the genome-wide extent.
- Targeted - utilised to capture specific genomic portions of particular interest.

Biosamples in this thesis represent WGS short-read data.

¹<https://www.pacb.com/blog/the-evolution-of-dna-sequencing-tools/>

2.2.1 NGS

The development of NGS methods started in the early 90s and was implemented in commercial DNA sequencers by 2000. It is denoted as massively parallel sequencing technology at elevated speeds while offering high throughput and scalability. NGS is used to determine the nucleic acid sequence in entire genomes or targeted regions of DNA or RNA by slicing the genome or its portions into fragments and randomly sampling the fragments (*shotgun* sequencing). Costs and speed are dependent on desired sequencing coverage.

2.3 DNase-seq

Deoxyribonuclease I (DNase I) is a term directly related to the DHSs and represents an enzyme that cuts DNA in an orderless manner [Hartmann, 2017]. It has been shown that active genes favour exhibiting altered nucleosome state [Weintraub and Groudine, 1976], making the DNase I an excellent tool for detecting and mapping specific genomic regulatory elements (DHSs). After the first discovery, the concept of DHSs and relevant DNase I studies drew much attention and peaked in the 1980s. Afterwards, it started to gradually decay, mainly because the traditional methods were not moderate to deliver significant results at a genome-wide scale. With the appearance of high-throughput technologies (e.g. NGS), the obstacles of conventional sequencing methods were overcome, and the revival of interest for DNase I digestion and DHSs started. Revival of interest led to the development of a sequencing method called DNase I hypersensitive site sequencing (DNase-seq) [Song and Crawford, 2010].

2 Related Work

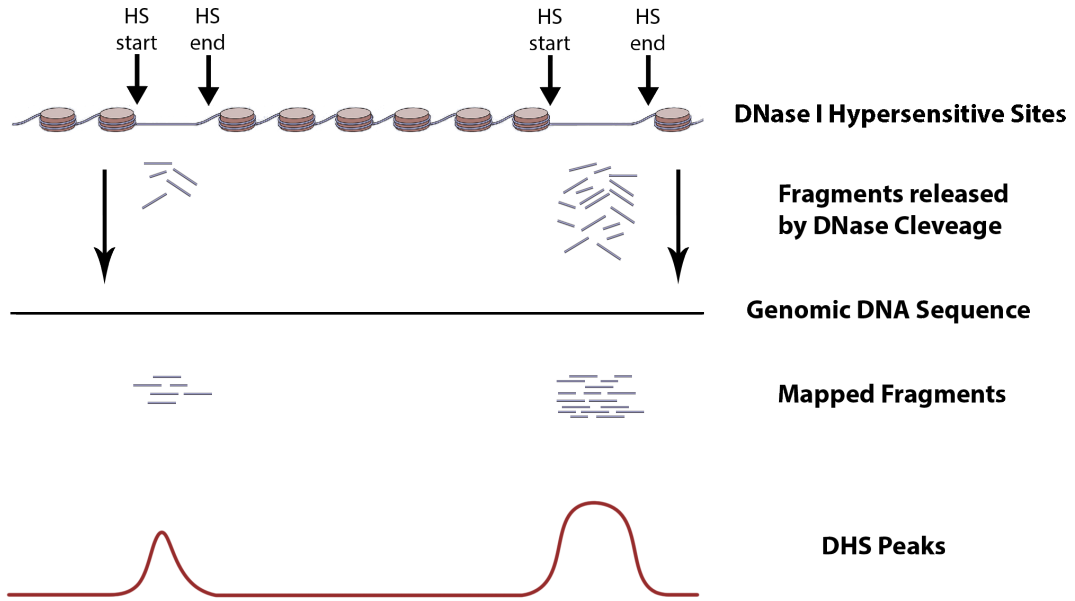


Figure 2.3: DNase-seq workflow. Detecting DHSs and building DHS genome-wide peak model for a biosample. Adapted from [Ling and Waxman, 2013]

The true novelty of DNase-seq is the potential to do the mapping of DNase I cleavage at nucleotide resolution on the genome-wide extent and improvement in signal-to-noise ratio (SNR), delivering clearer open chromatin signal. There are diverse applications of DNase-seq for extensive researches, varying from the investigation of nucleosome positions [Zhong et al., 2016] to the identification of genomic regions with nucleosome rotational stability [Winter et al., 2013] and the recognition of regulatory quantitative trait loci underlying expression variation [Degner et al., 2012]. Within the scope of this thesis, no DNase-seq was performed in our labs. Instead, already publicly available data has been utilised and subsequently analysed. This data was extracted and collected by external groups using DNase-seq with precisely chosen roadmaps, mainly as part of ENCODE 3 project [Moore et al., 2020].

2.4 Cell-free DNA

Circulating cell-free DNA (cfDNA) is a composite part of blood plasma and consists of highly degraded DNA fragments. Therefore, it has been intensively investigated as a biomarker and already proven that cfDNA released as a result of necrosis or apoptosis may have prognostic utility in various conditions like cancer, trauma, autoimmune diseases, cardiovascular disease, sepsis... [Heitzer, Haque, et al., 2019, Dwivedi2012Augff]. The idea is to capture these fragments before they get destroyed. Circulating cell-free DNA originating from tumour cells is called circulating tumour DNA (ctDNA).

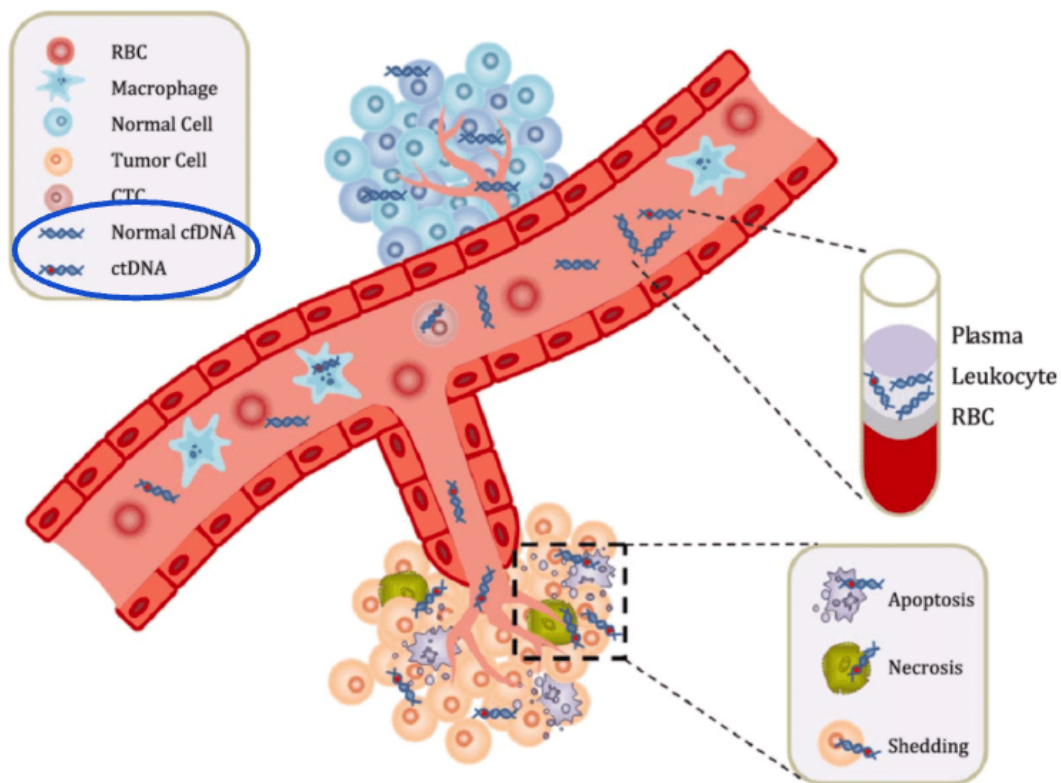


Figure 2.4: Blood plasma stream and constituting parts with the focus on cfDNA and ctDNA [Hahn et al., 2019]

2 Related Work

In order to detect cfDNA coming from a specific organ, reliance on organ-specific DNA methylation markers is required. The DNA fragmentation pattern is nonrandom, mainly due to its origin from apoptotic cells. In eukaryotic organisms (humans included), the chromosomal DNA is congested into a chain of perpetually repeating nucleosomes. Nonetheless, DNA directly associated with nucleosomes is protected from apoptosis and necrosis. Hence, cfDNA has a particular fragmentation pattern consisting of regions with different coverage patterns where high, and low coverage segments correspond to nucleosome protected and unprotected regions, respectively. The latter regions, i.e. the low coverage unprotected regions, correspond to open chromatin regions.

2.5 File Formats

2.5.1 FASTQ

The FASTQ file format is the mainstream textual format (human-readable) for storing biological sequences. As a result, it has been supported by more and more bioinformatics tools over time, making it perfectly suitable for our purposes. Compared to its predecessor, the FASTA², FASTQ contains quality information besides the sequence data. While both formats begin with a header line, the difference is that an @ character denotes the FASTQ header. For a single record (sequence read), there are four lines:

Line	Description
1	Always begins with '@' and then information about the read
2	The actual DNA sequence
3	Always begins with a '+' and sometimes the same info as in line 1
4	Has a string of characters which represent the quality scores; must have same number of characters as line 2

Figure 2.5: FASTQ file structure legend³

²<https://zhanggroup.org/FASTA/>

2 Related Work

Regarding the quality score, a mapping quality legend (Phred-33) looks like the following:

```
Quality encoding: !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHI
                  |           |           |           |           |
Quality score: 0.....10.....20.....30.....40
```

Figure 2.6: Phred-33 mapping quality legend⁴

Nucleotide (single base) call correctness is a probability model expressed through quality scores. Probability values result from the sequencer contained base-calling algorithm and are directly dependent on the strength of the captured signal. Quality scores are logarithm-based and calculated as:

$$Q = -10 \times \log_{10}(P)$$

Interpretation of Phred-33 quality scores is described in the following table:

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10000	99.99%

Figure 2.7: Phred-33 quality scores⁴

2.5.2 SAM/BAM

The BAM file format is becoming more and more popular among modern sequencing technologies. Its primary purpose is to store aligned sequences up to 128 Mb⁵. Therefore, it contains:

³<https://emea.support.illumina.com/bulletins/2016/04/fastq-files-explained.html>

⁴<https://learn.gencore.bio.nyu.edu/ngs-file-formats/quality-scores/>

⁵<https://samtools.github.io/hts-specs/SAMv1.pdf>

2 Related Work

- Header - contains information about the entire file, such as sample name, length and alignment method⁵.
- Alignments - contains read name, sequence, quality, alignment information and custom tags⁵.

BAM can express the same information as FASTQ plus mapping information in addition. The only difference is that the unaligned files are saved with the *unmapped* flag set to 1. This file format gained popularity because of its smaller disk footprint than the compressed FASTQ version. SAM is just text-based (human-readable) version of the BAM file.

2.5.3 BED

The BED file format is a text-based tab-separated format for storing genomic region coordinates and respective data. It was developed as a part of the *Human Genome Project*. With time it has been adapted by other sequencing projects. Although it has become a *de facto* standard in bioinformatics, it does not have official specifications. UCSC Genome Browser⁶ provides an extensive description and the file structure legend.

2.6 Biosample Datasets

We have a collection of biosamples, which have been occasionally sequenced in the scope of other projects over a couple of years. Biosamples used in this project correspond to WGS biosamples, mid to high-coverage ranging from ~10x to ~45x. Altogether, there are 116 biosamples in total encompassing the following three cohorts:

- CRC – Colorectal cancer – 37
- PC – Prostate cancer – 19
- HC – Healthy controls – 60

⁶<http://genome.cse.ucsc.edu/FAQ/FAQformat.html#format1>

2 Related Work

Eight biosamples (C2_6, C2_7, P40_1, P40_2, P147_1, P147_3, P148_1, P148_3) were sequenced on a half S4 NovaSeq flow cell with an average coverage of ~36x. The remaining biosamples were sequenced on a full S4 NovaSeq flow cell with an average coverage of ~15x.

2.6.1 Colorectal Cancer (CRC)

Ethical approval: 21-229 ex 09/10

CRC biosamples were collected as part of our routine liquid biopsy CRC cohort. Biosamples were selected for high-coverage WGS due to their high tumour fractions and interesting clinical features.

2.6.2 Prostate Cancer (PC)

Ethical approval: 21-228ex 09/10

PC biosamples were collected as part of our routine liquid biopsy PC cohort. Biosamples were selected for high-coverage WGS due to their high tumour fractions and interesting clinical features.

2.6.3 Healthy Controls

Sixty study participants were recruited. Study participants were between the age of 20 and 30 years. Half of them were female, and half were male. The complete blood count was collected for each individual to establish the total white blood cells (WBC) count and the broken down percentage of each WBC type along with hematopoietic progenitor cells. Study participants with known chronic or malignant diseases were excluded.

The following clinical data was collected from study participants:

- Age at blood collection
- Sex
- Smoking; alcohol consumption; current medication (if applicable);

2 Related Work

- History (family and own history): known diseases/hospital stays in the past and at the time of blood collection.

3 Methods

3.1 Processing of Biosamples

We have a collection of biosamples, which have been occasionally sequenced over a couple of years. Sequencing technology has changed with time, and so have the output files. Moreover, changes in the sequencing technology reflect deviations in the collected biosamples (e.g. parallel sequencing on four lanes instead of two, different adapter sequences to trim, output file format). Therefore, to handle the different scenarios and automate the uniform processing workflow, we develop a biosample processing pipeline called the *uBAM2FASTQ* pipeline. Pipeline workflow looks like following.

3 Methods

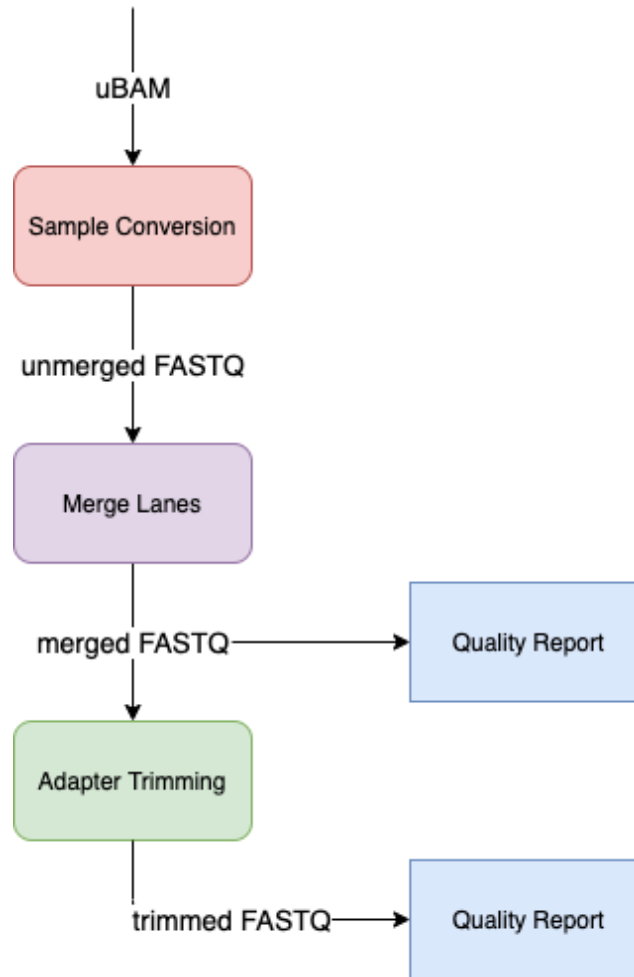


Figure 3.1: uBAM2FASTQ pipeline workflow

External tools like *cutadapt* and *samtools* already support multi-processing in contrast to *cat* and *FastQC*. Therefore, we manually implement multi-processing logic for *cat* and *FastQC*. In addition, we implement subprocess handling in all scripts to distribute data between child processes speeding up the processing of biosamples.

3.1.1 Conversion

We define *FASTQ* as the standard file format for our biosamples. *FASTQ* is a text-based (human-readable) format for storing biological sequences (Subsection 2.5.1). However, most of our collected biosamples are stored in the *uBAM* format. *uBAM* is a binary-based format for storing biological sequences (Subsection 2.5.2). Therefore, we perform conversion of *uBAM* biosamples to project defined standard format, the *FASTQ*. We utilise the *samtools fastq*¹ command for the conversion, passing respective input and output paths.

3.1.2 Merge Lanes

Modern DNA sequencers use multiple lanes to perform parallel sequencing. While parallel sequencing speeds up the sequencing process, it results in multiple output files of one biosample. Moreover, DNA sequencers produce parallel output such that the order of lanes corresponds to the merge order. Therefore, the merge process is intuitive and performed by concatenating lane files from 1 to N (the maximum number of lanes). We use the default Unix *cat* tool for the merge, passing the respective input and output paths. In addition, if the number of lanes is provided as an argument, we use it for error checking. The number of detected lanes may not be smaller or greater than the provided lane number. If not provided, the number of lanes has to be even.

3.1.3 Adapter Trimming

There is a common problem with adapters known as adapter contamination. It is present mainly in short-read sequencing. To solve the problem, we need to clean out the actual DNA fragments of adapters.

Adapter sequences are short, chemically synthesised, unique sequences. They can be ligated (attached) to both ends (5' and 3') of DNA fragments

¹<http://www.htslib.org/doc/samtools-fastq.html>

3 Methods

during the library preparation. These 5' and 3' sequences tag fragments and enable sequencers to detect them. Without those adapters attached to fragments, they would be undetectable to the sequencer.

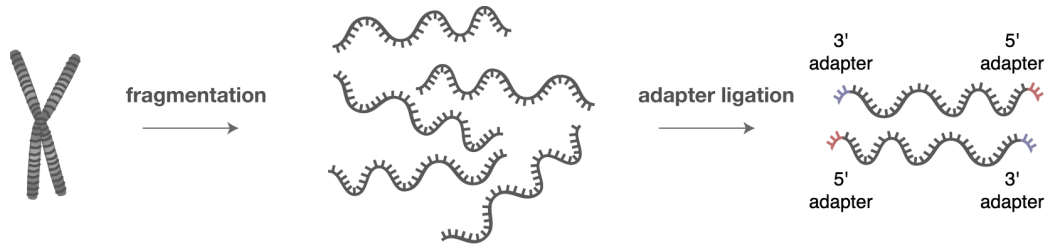


Figure 3.2: Fragmentation of a chromosome on fragments (molecules). Adapter ligation (attaching) to both ends (5' and 3') of fragments. Adapted from [Bioinformatics, 2016]

Adapter contamination occurs if the predefined sequencing (read) length exceeds the length of fragments themselves, causing the adapter sequence to get captured.

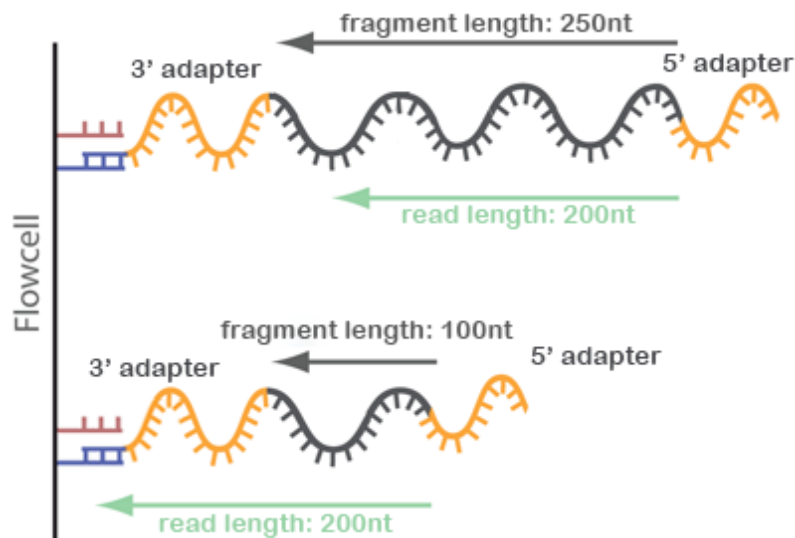


Figure 3.3: Read length shorter than the fragment length (top). Read length longer than the fragment length (bottom). Adapted from [Bioinformatics, 2016]

3 Methods

We need to identify the adapter sequence and trim it to recover the fragment. To identify the adapter sequence (e.g. Illumina Universal Adapter), we exploit information from the quality report (Subsection 3.1.4). We utilise a tool called *cutadapt* (version 3.4) for the trimming process, which supports multi-processing.

3.1.4 Quality Report

Errors occur during sequencing and library preparation, leading to corrupt patient data. Quality reports are created and assessed to provide insight into biosamples. We use them as filter criteria to distinguish between corrupt and uncorrupt biosamples and provide an overview of specific biosample quality metrics that could affect future analyses (e.g. coverage signals).

For the quality report generation, we utilise a tool called *FastQC*. This tool produces quality metrics of input biosample as an HTML file. It enables us to conduct a modular set of analyses providing a quick impression of whether the data exhibits any problems. Furthermore, we inspect these metrics to decide if we should exclude, reevaluate or include biosamples in the analysis. For example, if a biosample is classified as **Failure** in any of the following metrics, we would exclude it.

Metrics are expressed in quality scores (Figure 2.7). Among all metrics contained in the report, following are of particular interest to us:

- Per Base Sequence Quality - X-axis depicts read bases. Y-axis depicts quality scores. The higher the score, the better the base call. Y-axis displays three different colours based on the call quality: good quality calls (green), reasonable quality calls (orange), and poor quality calls (red)².
- Per Sequence Quality Scores - This module allows seeing if a subset of sequences has low-quality values. A subset of sequences will often have universally poor quality, mainly because they are poorly imaged (e.g. on the edge of the field of view); however, these should represent only a small percentage of the total sequences².

²<https://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/3%20Analysis%20Modules/>

3 Methods

- Per Base Sequence Content - This module plots the proportion of each base position in a file for which each of the four standard DNA bases has been called. If one distribution of the bases deviates to a large extent from the others, it will be classified as **Failure**².
- Per Base N Content - If the sequencer cannot make a base call with sufficient confidence, it will typically substitute an N with a conventional base call. This module plots the percentage of base calls at each position for which an N was called².
- Sequence Length Distribution - Some sequencers generate sequence fragments (reads) of uniform length, but others can contain reads of wildly varying lengths. This module generates a graph showing the distribution of fragment sizes in the input file².
- Adapter Content - The plot shows a cumulative percentage count of the proportion of detected adapter sequences in all sequenced reads².

All of our biosamples pass the quality checks.

3.2 Clustering of DNase-seq Data

DNase-seq is a sequencing method used to capture DHSs (Section 2.3). DHSs represent a specific group of genomic regions related to cancer biology.

To compose as specific DHS datasets as possible targeting different cancer types, we exploit DNase-seq data. However, DHSs detected using DNase-seq exhibit high variability in terms of position between different biosamples (due to biological differences). They are shifted somewhat to the left or right in different biosamples. Also, there is a case when specific DHS is completely missing in a biosample. In an ideal scenario, all the detected DHSs would be in the same position in all biosamples, and none would be missing.

To overcome the addressed challenges, we come up with two methods:

- Custom Clustering
- Bedtools-based Clustering

3 Methods

In the following, we utilise the data presented in **Vierstra et al., 2020**. The procedure can be easily extended to any other DNase-seq data, as long as some tissue or cell-line specificity is bound to it.

3.2.1 Retrieval of Vierstra DNase-seq Data for Clustering

Searching for DHS datasets, we come across DNase-seq data grouped according to target organs/tissues. Data is presented in **Vierstra et al., 2020**. We aim to create tissue-specific DHS datasets targeting different cancer types, thus targeting different biosample cohorts. Therefore, realising the great potential of the data, we utilise it to create custom DHS datasets.

The data is presented in form of *Human Body Map*.

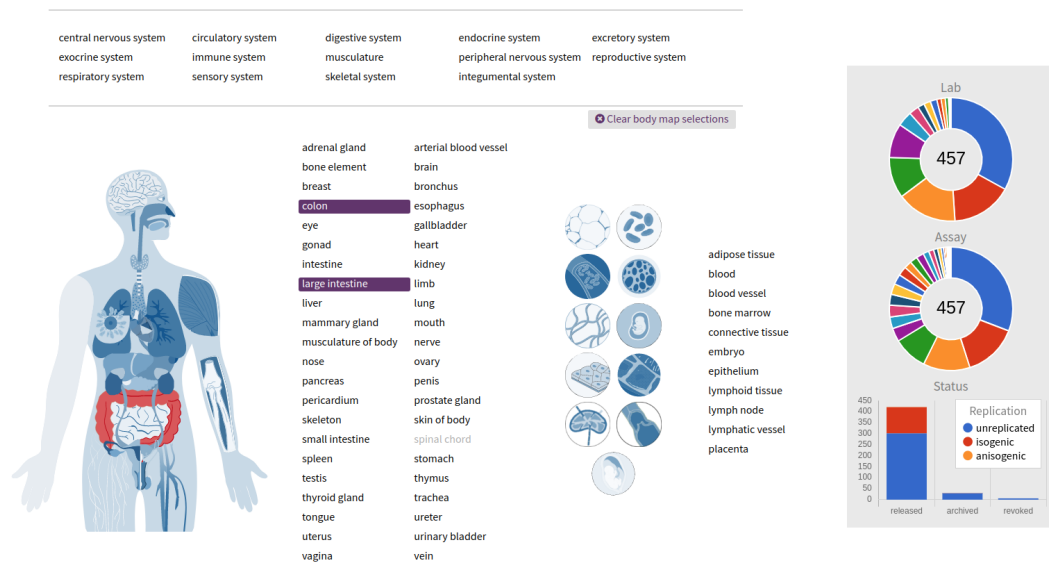


Figure 3.4: Interface of *Human Body Map*. Colon and large intestine organs selected³

³https://www.encodeproject.org/summary/?type=Experiment&replicates.library.biosample.donor.organism.scientific_name=Homo+sapiens&biosample_ontology.organ_slims=colon&biosample_ontology.organ_slims=large+intestine#openModal

3 Methods

As the first example, we opt to create custom *digestive* dataset by picking *large intestine* and *colon* as the target tissues. We aim to distinguish CRC biosamples from other biosample cohorts by targeting the *digestive* system. However, this approach can be easily extended to any other target tissue to create other tissue-specific DHS datasets.

Human Body Map contains sequencing data from various sequencing methods. Nonetheless, the *DNase-seq* column is of particular interest to us.

← BIOSAMPLE	ASSAY →																												
	Histone ChIP-seq	TF ChIP-seq	Control ChIP-seq	DNase-seq	total RNA-seq	DNAmne array	WGS	ATAC-seq	Hi-C	polyA plus RNA-seq	RNA microarray	RAMPAGE	RRBS	small RNA-seq	microRNA-seq	WGBS	genotyping array	long read RNA-seq	ChIA-PET	Bru-seq	BruChase-seq	5C	BruUV-seq	FAIRE-seq	MRE-seq	microRNA counts			
▼ tissue	119	37	47	33	14	11	14	12	4	11	9	8	7	8	5	6	4	3							1				
sigmoid colon	35	19	22	5	4	4		4		4		4		4		2													
transverse colon	22	18	14	7	4	4	14	4	4			4		4		2	4												
large intestine	6		1	15						7	5					1													
colonic mucosa	20		3	1	2			1			1		2		1	1													
mucosa of rectum	14		2								2		2																
▼ cell line	22	28	9	8	2	3		1	9	1	2	1	2	1	2		1	1	3	2	2	1	1		1	1	1		
HCT116	17	27	7	1	1	2		1	8	1	1		1		2			1	3	2	2		1		1	1	1		
Caco-2	5	1	2	2		1					1		1				1												
HT-29				1	1						1			1								1							
DLD1								1						1															
HCEC 1CT				1																									

Figure 3.5: *Human Body Map* displaying available sequencing data for selected colon and large intestine⁴. DNase-seq column is highlighted

Conducting the described steps, we retrieve 33 tissue-based and 8 cell line-based DNase-seq datasets. To retain only high-quality data, we filter out biosamples relating to any of the following points:

- Extremely low SPOT score - The SPOT (Signal Portion of Tags) score is a metric for the quality of DNase-seq data. Biosamples with extremely low SPOT score are not trustable.

⁴https://www.encodeproject.org/matrix/?type=Experiment&replicates.library.biosample.donor.organism.scientific_name=Homo+sapiens&biosample_ontology.organ_slims=colon&biosample_ontology.organ_slims=large+intestine

3 Methods

- Missing footprints - Analysis was unable to define DNase-seq footprints confidently.
- Revoked
- Archived
- Embryo biosamples

We end up with 19 biosamples, whereas 11 stem from primary tissue and 8 stem from specific cell types. So eventually, two separate clustering procedures have to be conducted—one for primary tissue and one for cell types.

Downloaded files are in BED format (Subsection 2.5.3) and contain DNase-seq detected DHSs.

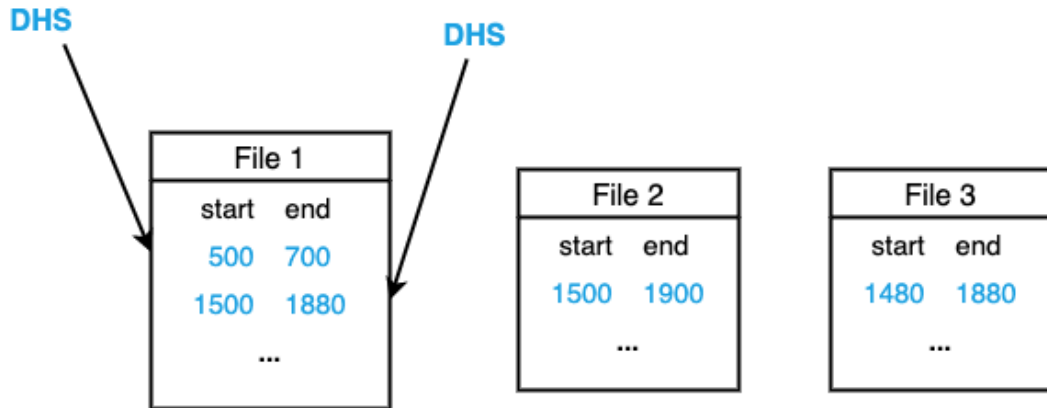


Figure 3.6: Example of DNase-seq output file structure

3.2.2 DNase-seq Data Custom Clustering Method

Due to human nature, DHSs differ between biosamples in terms of genomic location or presence in general thus, no perfect DHS overlap is possible. In the following, we present a method we developed to overcome this challenge and cluster associated or functionally congruent DHSs. First, we merge all downloaded DNase-seq files of a specific tissue/organ and then sort according to the DHS *start* and *end* positions.

3 Methods

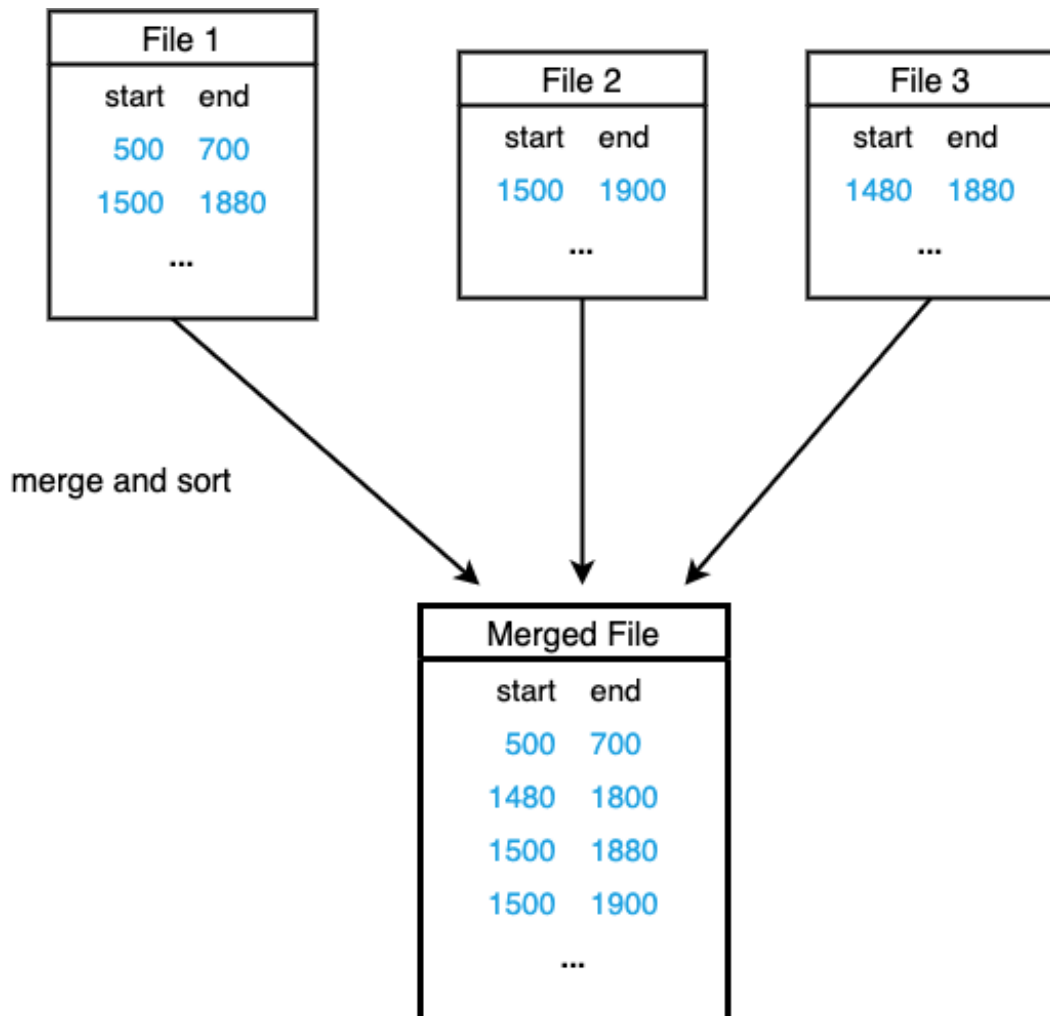


Figure 3.7: Merge and sort of DNase-seq files according to their start and end position

Representing the merged file on a portion of the genome, as depicted in Figure 3.8, we can straightforwardly elaborate on the clustering condition. The first DHS of every cluster is taken as the reference. Then, it is compared to all following DHSs until the break condition is met. The brake condition is met when one DHS has its *start* position larger than the *end* position of reference DHS. DHS fulfilling the brake condition is then taken as the reference DHS for the next cluster, and the process is repeated until there

3 Methods

are no more DHSs.

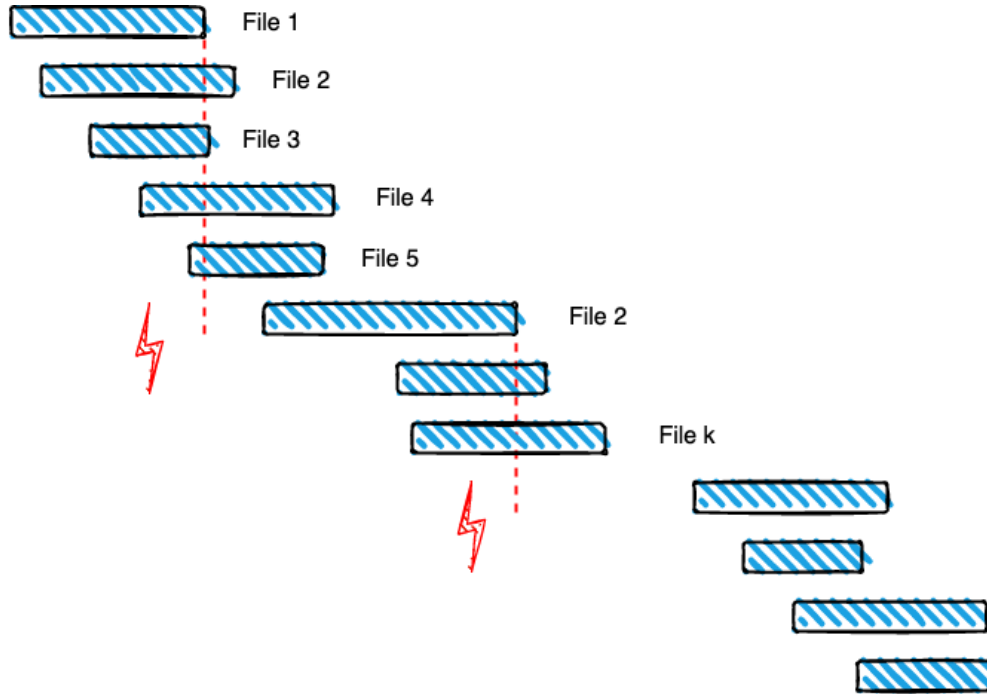


Figure 3.8: 2D representation of DHSs on a portion of genome. Red lines express brake condition for *Custom Clustering* method

During cluster creation, we retain the following metrics:

- start
- end
- core_start
- core_end
- overlap percentage

Metrics *start* and *end* respectively represent the *start* position of the *first DHS* in the cluster and the *end* position of the *last DHS* in the cluster. Likewise, metrics *core_start* and *core_end* represent the *start* position of the *last DHS* in

3 Methods

a cluster and the *end* position of the *first* DHS in a cluster. Finally, *overlap percentage* provides percentwise insight into the number of overlapping DHSs in the cluster. This metric is used as a filter for the strictness of clusters. If a cluster does not satisfy desired strictness, we discard it.

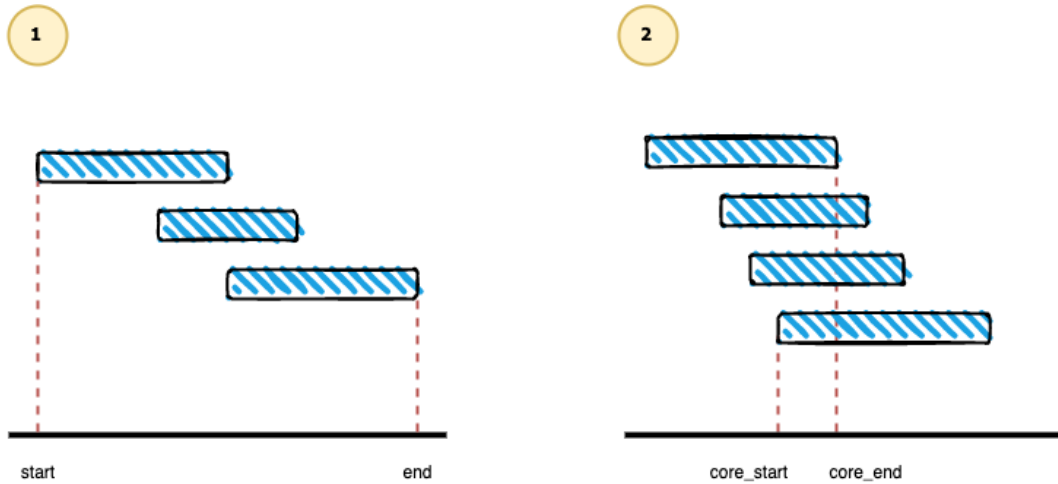


Figure 3.9: 2D demonstration how the four cluster metrics (*start*, *end*, *core_start*, *core_end*) are calculated

3.2.3 DNase-seq Data Bedtools-based Clustering Method

We implement an additional method to perform the clustering of DHSs. The main motivation behind this method is to attain a benchmark factor and test for improvement regarding the custom clustering method (Subsection 3.2.2).

We conduct this clustering method using an external tool bundle named *bedtools*⁵ (version 2.30.0). As the name indicates, it supports the processing of BED files, which is precisely the format of downloaded DNase-seq data. Therefore, we utilise the *bedtools multiinter*⁶ command to retrieve intersecting DHS regions in the DNase-seq data. This tool contains no clustering logic

⁵<https://bedtools.readthedocs.io/en/latest/>

⁶<https://bedtools.readthedocs.io/en/latest/content/tools/multiinter.html>

3 Methods

whatsoever, but instead, it just performs position-wise (per base) intersection of contained DHSs inside the file.

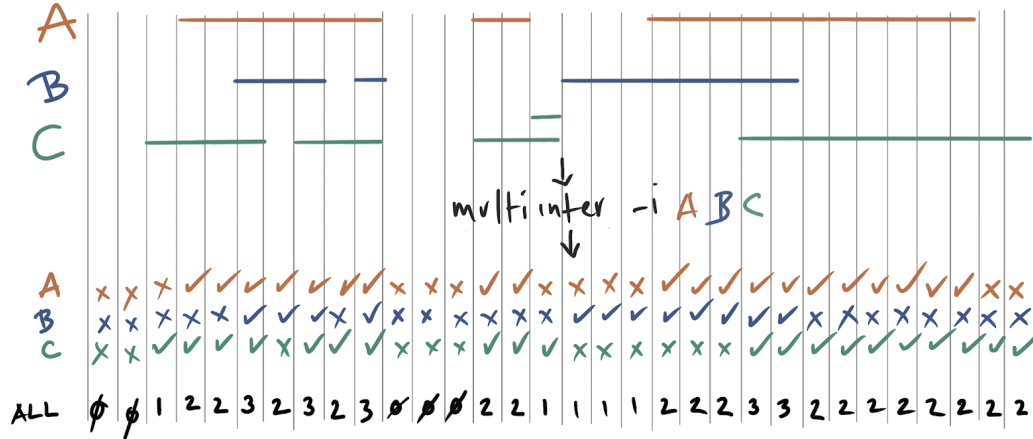


Figure 3.10: 2D demonstration how *bedtools multiinter* calculates overlaps⁷

It produces output as genomic positions along with the respective number of intersecting files at those positions.

```
$ bedtools multiinter -header -names A B C -i a.bed b.bed c.bed
chrom    start  end    num    list    A    B    C
chr1      6      8      1      1      1    0    0
chr1      8     12      2     1,3    1    0    1
chr1     12     15      3     1,2,3  1    1    1
chr1     15     20      2     1,2    1    1    0
chr1     20     22      1      2      0    1    0
chr1     22     30      2     1,2    1    1    0
chr1     30     32      1      2      0    1    0
chr1     32     34      1      3      0    0    1
```

Figure 3.11: Example of *bedtools multiinter* output format

Then, we develop a script that extracts clusters from this output file regarding the desired cluster strictness. The intersection percentage was computed by dividing the number from *num* column by the total number of input files. All genomic positions whose intersection percentage satisfies the desired

⁷<https://bedtools.readthedocs.io/en/latest/content/tools/multiinter.html>

strictness are taken into account. Each continuous sequence (more than one) of positions represents a cluster. Moreover, the position of the first intersection represents the *start*, and the last intersection represents the *end* of the corresponding cluster.

3.3 Processing of Pre-clustered DHS Datasets

Pre-clustered DHSs are already established representative DHSs. Furthermore, the dataset authors performed their clustering method along with DNase-seq. Therefore, there is no need for own clustering methods. The goal is to find tissue/organ-specific DHS datasets, thereby targeting different cancer types.

Searching for DHS datasets, we came across two valuable sources of pre-clustered DHSs published in **Sheffield et al., 2013** and **Meuleman et al., 2020**. Realising the great potential and high-specificity of the data, we decide to exploit it. Although both of these publications provide pre-clustered DHSs, the data slightly differs. Therefore, a certain degree of processing is required depending on the degree of freedom left by the author.

3.3.1 Collecting and Lifting-over Sheffield Duke Clusters

Data provided in **Sheffield et al., 2013** comes in tissue/cell line-specific format. Therefore, we select individually tissues/cell lines targeting respective cancer types of our biosample cohorts. For that purpose, we exploit the regulatory elements database⁸ and pick the most significant DHS clusters for specific tissues based on belonging cell lines.

⁸<http://dnase.genome.duke.edu/index.php>

3 Methods

Choose samples to include:

Toggle All

Clear All

Submit

Epithelial	Fibroblast	Muscle	Brain	Colon	Hematopoietic	Primitive	Skin	Stem	Endothelial	Cervix	Liver	Prostate	Mammary	Bone
Group	Group	Group	Group	Group	Group	Group	Group	Group	Group	Group	Group	Group	Group	Group
<input type="checkbox"/> A549	<input type="checkbox"/> AG04449	<input type="checkbox"/> AG04450	<input type="checkbox"/> BE2_C	<input type="checkbox"/> HCT-116	<input type="checkbox"/> CD14	<input type="checkbox"/> CLL	<input type="checkbox"/> Chorion	<input type="checkbox"/> Colo829	<input type="checkbox"/> HBMEC	<input type="checkbox"/> HeLa-S3	<input type="checkbox"/> Hepatocytes	<input type="checkbox"/> LNCaP	<input type="checkbox"/> MCF-7	<input type="checkbox"/> Osteoblast
<input type="checkbox"/> HAEpIC	<input type="checkbox"/> AG09309	<input type="checkbox"/> AoSMC_SF	<input type="checkbox"/> Glioblastoma	<input type="checkbox"/> CMK	<input type="checkbox"/> GM06990	<input type="checkbox"/> GM12864	<input type="checkbox"/> Htr8	<input type="checkbox"/> hESC	<input type="checkbox"/> HMVEC-dBI-Ad	<input type="checkbox"/> HeLa-S3_IFNA	<input type="checkbox"/> HepG2	<input type="checkbox"/> LNCaP_andro	<input type="checkbox"/> MCF-7_hyp_lac	
<input type="checkbox"/> HCPEpIC	<input type="checkbox"/> AG09319	<input type="checkbox"/> HCM	<input type="checkbox"/> HA-c	<input type="checkbox"/> GM12865	<input type="checkbox"/> GM12878	<input type="checkbox"/> GM12891	<input type="checkbox"/> Melano	<input type="checkbox"/> hESC	<input type="checkbox"/> HMVEC-dBI-Neo	<input type="checkbox"/> Huh-7	<input type="checkbox"/> Huh-75			
<input type="checkbox"/> HEEpIC	<input type="checkbox"/> AG10803	<input type="checkbox"/> HSM	<input type="checkbox"/> HA-sp	<input type="checkbox"/> GM12892	<input type="checkbox"/> GM18507	<input type="checkbox"/> GM19238	<input type="checkbox"/> NHEK	<input type="checkbox"/> hESC	<input type="checkbox"/> HMVEC-dLy-Ad	<input type="checkbox"/> PA-TU-898T				
<input type="checkbox"/> HIPEpIC	<input type="checkbox"/> AoAF	<input type="checkbox"/> HSMtube	<input type="checkbox"/> HAh	<input type="checkbox"/> GM19239	<input type="checkbox"/> GM19240	<input type="checkbox"/> HL-60	<input type="checkbox"/> IPS	<input type="checkbox"/> hESC	<input type="checkbox"/> HMVEC-dLy-Neo					
<input type="checkbox"/> HMEC	<input type="checkbox"/> BJ	<input type="checkbox"/> Myometr	<input type="checkbox"/> Medullo	<input type="checkbox"/> Jurkat	<input type="checkbox"/> K562			<input type="checkbox"/> Ntera2	<input type="checkbox"/> HMVEC-dNeo					
<input type="checkbox"/> HNPEpIC	<input type="checkbox"/> SKMC	<input type="checkbox"/> NHA	<input type="checkbox"/> SK-N-SH_RA	<input type="checkbox"/> Th1					<input type="checkbox"/> HMVEC-LBI					
<input type="checkbox"/> HPDE6-E6E7	<input type="checkbox"/> Fibroblast		<input type="checkbox"/> SK-N-SH_RA	<input type="checkbox"/> Th2					<input type="checkbox"/> HMVEC-LLy					
<input type="checkbox"/> HRCE	<input type="checkbox"/> FibroP		<input type="checkbox"/> SKNMC						<input type="checkbox"/> HMVECdAd					
<input type="checkbox"/> HRE	<input type="checkbox"/> HCF								<input type="checkbox"/> HPAEC					
<input type="checkbox"/> HRPEpIC	<input type="checkbox"/> HCFaa								<input type="checkbox"/> HRGEC					
<input type="checkbox"/> PANC-1	<input type="checkbox"/> HConF								<input type="checkbox"/> HUVEC					
<input type="checkbox"/> PREC	<input type="checkbox"/> HFF													
<input type="checkbox"/> RPTEC	<input type="checkbox"/> HFF_Myc													
<input type="checkbox"/> SAEC	<input type="checkbox"/> HGF													
<input type="checkbox"/>	<input type="checkbox"/> HMF													
<input type="checkbox"/> Urothelia_UT189	<input type="checkbox"/> HPAF													
	<input type="checkbox"/> HPdLF													
	<input type="checkbox"/> HPF													
	<input type="checkbox"/> HVMF													
	<input type="checkbox"/> NHDF-Ad													
	<input type="checkbox"/> NHDF-neo													
	<input type="checkbox"/> NHLF													
	<input type="checkbox"/> ProgFib													
	<input type="checkbox"/> Stellate													
	<input type="checkbox"/> WI-38													
	<input type="checkbox"/> WI-38-TAM													

Choose samples to exclude:

Toggle All

Clear All

Submit

Figure 3.12: Interface for Duke DHS Cluster Database⁹ depicting tissue type as column name and belonging cell lines as its rows

34

3 Methods

The green area checkboxes represent cell lines whose DHSs are to be included, and the red area checkboxes cell lines whose DHSs are to be excluded. If a checkbox is checked in neither green nor red area, clusters will be retrieved regardless of whether they contain these cell lines or not.

After being presented with DHS clusters, we look for ones that exhibit adequate tissue-specific accessibility patterns. Adequate pattern reflects high accessibility values for belonging cell lines and low for non-belongs cell lines. Accessibility value defines how open (accessible) are belonging DHSs based on their DNase-seq data.

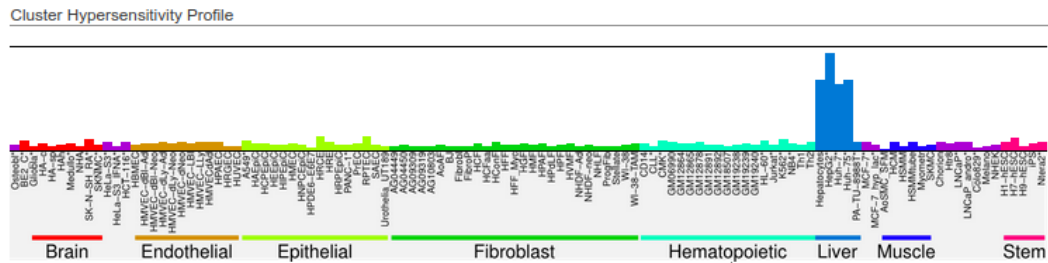


Figure 3.13: Liver cluster 1066¹⁰ with an adequate tissue-specific accessibility pattern (blue bars in comparison to others)

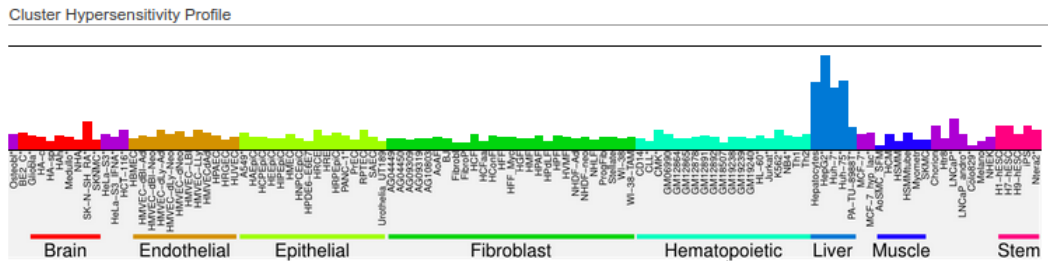


Figure 3.14: Liver cluster 1115¹¹ with an adequate tissue-specific accessibility pattern (blue bars in comparison to others)

⁹<http://dnase.genome.duke.edu/celltype.php>

¹⁰<http://dnase.genome.duke.edu/clusterDetail.php?clusterID=1066>

¹¹<http://dnase.genome.duke.edu/clusterDetail.php?clusterID=1115>

3 Methods

All the clusters in the following are based on GRCh37 (hg19) reference genome build, which happened to be the latest version at the time of publication. Therefore, we convert their genomic coordinates to the up-to-date reference genome build, the GRCh38 (hg38). The process of converting genomic coordinates between two reference genome builds is called **lift-over**. For the genomic coordinate conversion between these two reference genome builds, we use UCSC liftOver tool¹². **Lift-over** results in an insignificant loss of 0-3% DHSs depending on the cluster.

Prostate-specific Clusters

We select prostate-specific clusters by including *PrEC* and excluding everything but the *Epithelial* column.

We retrieve 6 clusters. After analysing tissue-specific accessibility patterns, we pick out 4 clusters:

- ID 1982 (#235 sites)
- ID 2100 (#2275 sites)
- ID 2150 (#4650 sites)
- ID 2357 (#664 sites)

Prostate Cancer-specific Clusters

We select prostate-specific clusters by including both *LNCaP* and *LNCaP_andro* from the Prostate column and excluding everything but the *Prostate* column.

We retrieve only 1 cluster with a relatively good tissue-specific accessibility pattern:

- ID 2483 (#1135 sites)

¹²<https://genome.ucsc.edu/cgi-bin/hgLiftOver>

Hematopoietic Clusters

We select hematopoietic clusters by including *GM06990*, *GM12864*, *GM12865*, *GM12878*, *GM12891*, *GM12892*, *GM18507*, *GM19238*, *GM19239*, *GM19240* together and excluding everything but the *Hematopoietic* column.

We retrieve 22 clusters. After analysing tissue-specific accessibility patterns, we pick out 10 clusters:

- ID 22 (#777 sites)
- ID 23 (#242 sites)
- ID 24 (#201 sites)
- ID 25 (#270 sites)
- ID 72 (#326 sites)
- ID 73 (#181 sites)
- ID 123 (#261 sites)
- ID 125 (#191 sites)
- ID 275 (#200 sites)
- ID 765 (#189 sites)

Epithelial Clusters

Since the *Epithelial* is vast in terms of originating tissues because it contains cell lines belonging to different cancer types. We build subgroups of clusters by including the corresponding cell lines individually.

We include *HEEpiC*, *A549*, *HRCE*, *PANC-1* individually and exclude everything but the *Epithelial* column.

For the *HEEpiC* subgroup, we retrieve 6 clusters, whereas 4 of them have good tissue-specific accessibility patterns:

- ID 1982 (#235 sites)
- ID 2100 (#2275 sites)
- ID 2150 (#4650 sites)
- ID 2357 (#664 sites)

For the *A549* subgroup, we retrieve 6 clusters, whereas 2 of them have good tissue-specific accessibility patterns:

3 Methods

- ID 2076 (#2076 sites)
- ID 2460 (#337 sites)

For the *HRCE* subgroup, we retrieve 9 clusters, whereas 1 of them has a good tissue-specific accessibility pattern:

- ID 1766 (#311 sites)

For the *PANC-1* subgroup, we retrieve 7 clusters, whereas 2 of them have good tissue-specific accessibility pattern:

- ID 773 (#2004 sites)
- ID 1974 (#3912 sites)
- ID 24 (#201 sites)
- ID 25 (#270 sites)
- ID 72 (#326 sites)
- ID 73 (#181 sites)
- ID 123 (#261 sites)
- ID 125 (#191 sites)
- ID 275 (#200 sites)
- ID 765 (#189 sites)

PANC-1 is supposed to show difference in CRC biosamples in comparison to other biosamples, since it is a *digestive* dataset.

Liver Clusters

We select liver-specific clusters by including *Hepatocytes* from the *Liver* column and excluding everything but the *Liver* column. We retrieve 14 clusters. After analysing tissue-specific accessibility patterns, we pick out 2 clusters:

- ID 1066 (#437 sites)
- ID 1115 (#265 sites)

3.3.2 Statistical Preprocessing of Meuleman Intermediate DHS Matrix

Meuleman intermediate DHS matrix contains accessibility values of ~3,6 million (pre-clustered DNase-seq) DHSs across 733 biosamples. In this case, accessibility value defines how open is a particular DHS in a particular biosample.

Biosamples carry biological differences, making raw accessibility values uncomparable between different biosamples. In order to mitigate biological biases and make the accessibility values comparable, we conduct specific statistical preprocessing steps in the exact order:

1. Winsorization
2. Quantile Normalisation
3. MinMax Scaling

For the sake of visualisation, the following plots display only 20 biosamples out of 733. We use boxplots to display distributions of accessibility values on the biosample level. They are also more accurate than kernel density estimation (KDE) plots. Although less accurate representations of this data, KDE plots are used as an alternative to boxplots to provide insight into a different perspective of overlapping distributions.

Looking at the boxplot, we can observe that most values are distributed around 0, but there are certain biosamples whose values outlie even to values larger than 600.

3 Methods

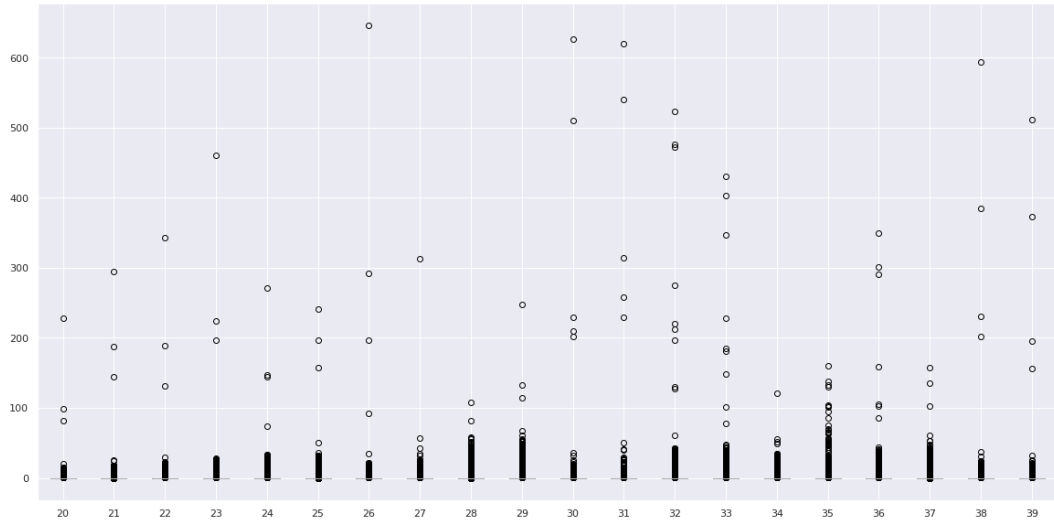


Figure 3.15: Boxplot of Meuleman DHS matrix biosamples (20 - 40) and their raw DHS accessibility values

Looking at the KDE plot, we can observe that most biosamples have their distribution accumulated at 0. However, we have no information about the outliers and from which biosample they originate. That is the lack of information mentioned previously about KDE.

3 Methods

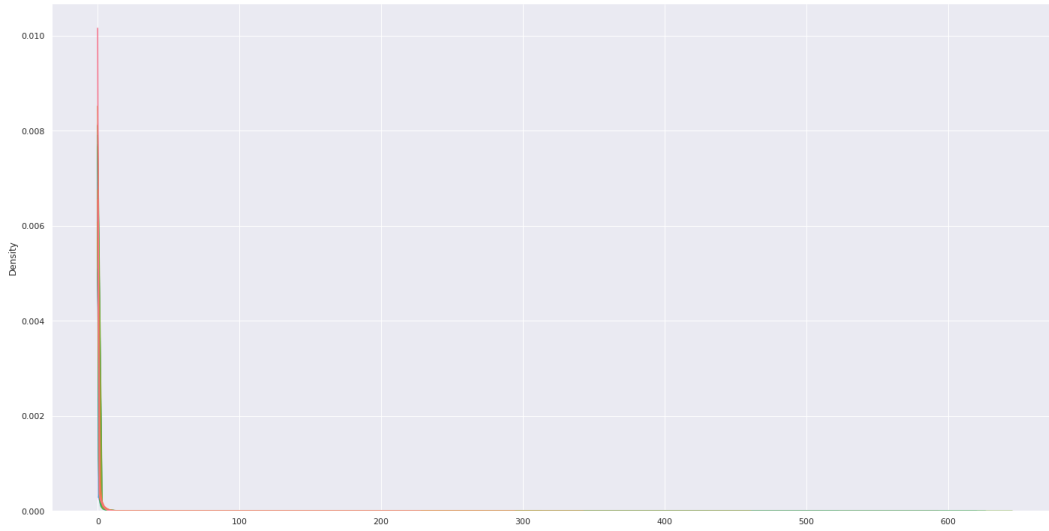


Figure 3.16: KDE of Meuleman DHS matrix biosamples (20 - 40) and their raw DHS accessibility values

Winsorization

We perform winsorization to remove spurious outliers and balance the accessibility values within a biosample. First, we conduct winsorization of 5% (2.5% from both sides of distribution). It scales the values below the 2.5th percentile to the 2.5th percentile and values above the 97.5th percentile to the 97.5th percentile.

Looking at the boxplot, we can observe that many points beyond individual whiskers got rescaled to the area around 0. This implies that most accessibility values are around 0, but the plot is biased due to spurious outliers.

3 Methods

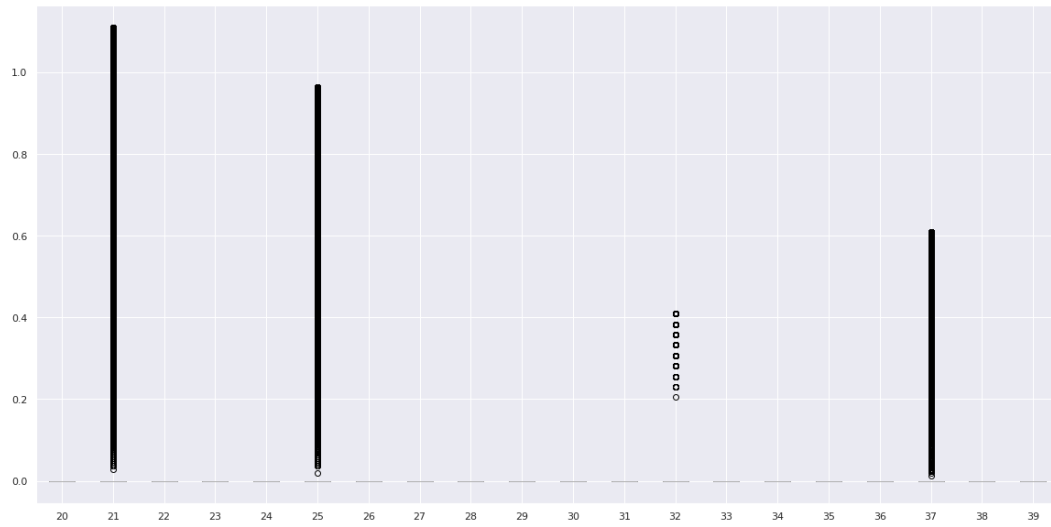


Figure 3.17: Boxplot of Meuleman DHS matrix biosamples (20 - 40) and their winsorized DHS accessibility values by 5%

The KDE plot shows peaks beyond 0 on x-axis. These peaks correspond to the highest outlying point for respective biosamples (e.g. ~ 0.4 , ~ 0.41 , ~ 0.61 , ~ 0.9 , ~ 1.5).

3 Methods

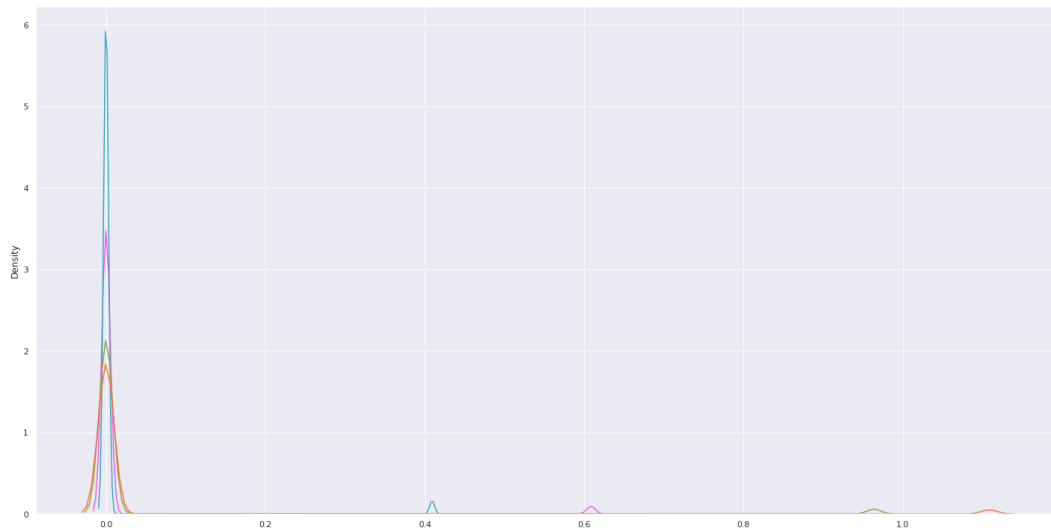


Figure 3.18: KDE of Meuleman DHS matrix biosamples (20 - 40) and their winsorized DHS accessibility values by 5%

Quantile Normalisation

We perform quantile normalisation to make the accessibility value distributions between biosamples identical in statistical properties. There is no reference distribution to normalise the data, but the accessibility value distributions are normalised to each other.

Performing quantile normalisation on these biosamples, their accessibility values get slightly rescaled, and their whiskers seem to be at the same level as other biosamples. After the values got rescaled, it is now clearly visible that most accessibility scores are not exactly at 0 but slightly above it.

3 Methods

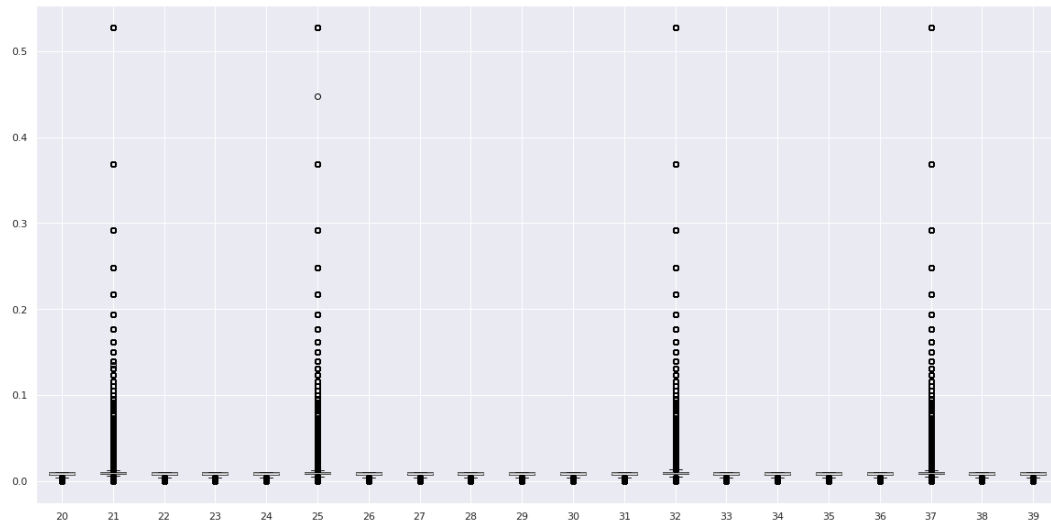


Figure 3.19: Boxplot of Meuleman DHS matrix biosamples (20 - 40) DHS accessibility values. Winsorization (5%) and quantile normalisation of accessibility value distributions to each other applied

Looking at the Figure 3.20, we can observe that many more distributions are now overlapping at 0 what corresponds to the boxplot previously shown. The sudden peaks are also gone, since the distributions are stretched out to be on the same scale. Therefore, their distributions became comparable to each other.

3 Methods

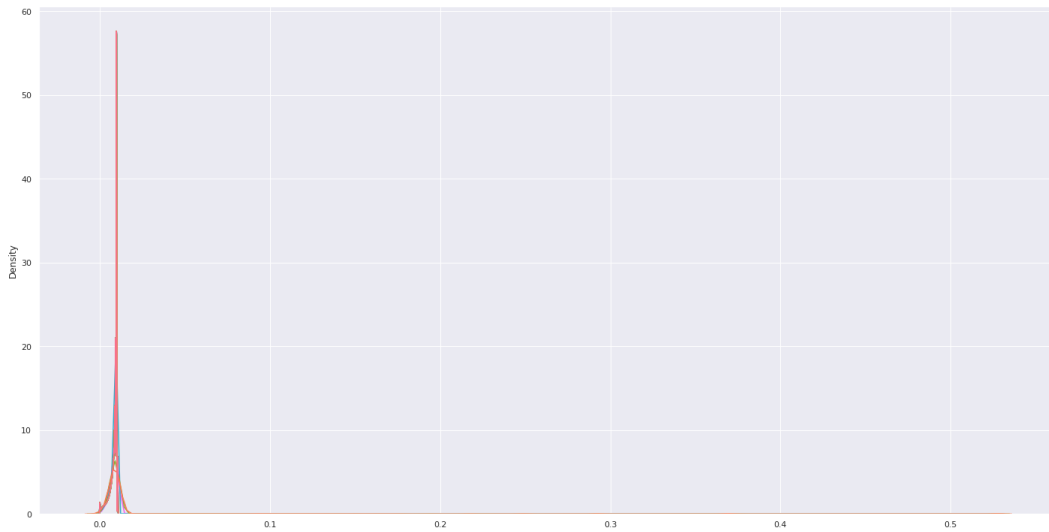


Figure 3.20: KDE of Meuleman DHS matrix biosamples (20 - 40) DHS accessibility values. Winsorization (5%) and quantile normalisation of accessibility value distributions to each other applied

MinMax Scaling

Some accessibility values could stretch out tremendously due to quantile normalisation, which would skew the underlying distribution. In order to reduce the effect, we perform MinMax scaling between 0 and 1.

After performing MinMax scaling, we can clearly observe the accessibility value differences between biosamples. For example, some DHSs exhibit low accessibility values due to biological differences compared to the biosamples with high accessibility values.

3 Methods

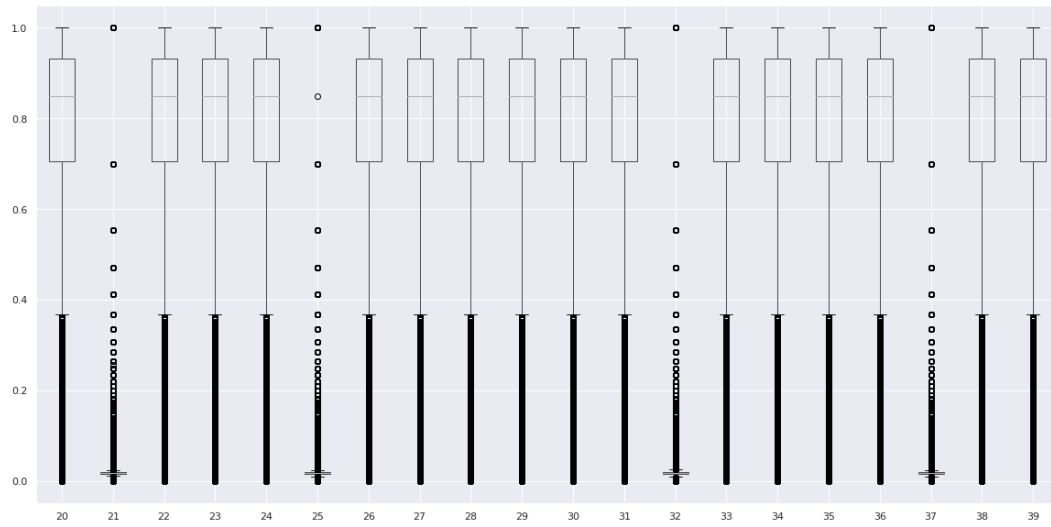


Figure 3.21: Boxplot of Meuleman DHS matrix biosamples (20 - 40) DHS accessibility values. Winsorization (5%), quantile normalisation and MinMax scaling between 0 and 1 of accessibility values applied

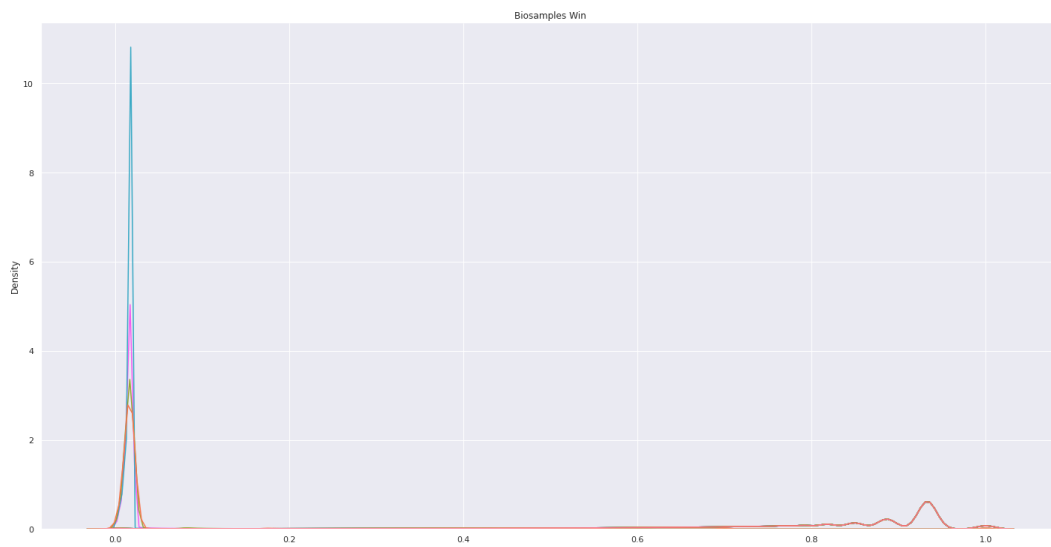


Figure 3.22: KDE of Meuleman DHS matrix biosamples (20 - 40) DHS accessibility values. Winsorization (5%), quantile normalisation and MinMax scaling between 0 and 1 of accessibility values applied

Filter Criteria

After setting a common scale for accessibility scores, there is a demand for building filter criteria. Not all DHSs are equally important and they cannot be filtered by picking DHSs with high accessibility values in biosamples with a specific disease. By doing so, there will be many DHSs with high accessibility values in both healthy and diseased biosamples. Therefore, we built the following filter:

1. Pick group of biosamples with a specific disease
2. At most 5% of the non-picked out biosamples have accessibility values above 0.2
3. The most accessible non-picked out biosample has a value below 0.5
4. Sort remaining DHSs by their median accessibility in the picked out biosamples and use top 1k

3.3.3 Splitting Meuleman DHS Components

Meuleman DHS Components file contains ~3,6 million representative (pre-clustered DNase-seq) DHSs along with predicted *components*. Moreover, a component defines tissue specificity.

Each row in the file represents one DHS belonging to one of the sixteen **components** with no information about origin biosamples. The header structure is shown in following figure.

3 Methods

Column	Example	Description
seqname	chr1	Chromosome
start	1782520	Start position
end	1782770	End position
identifier	1.10643	Unique identifier (chr#.position%)
mean_signal	1.030869481	Mean DNase-seq signal across biosamples with a DHS ("confidence score")
numsamples	54	Number of biosamples with a DHS
summit	1782650	Estimated DHS summit position
core_start	1782590	Start position of core-region containing 95% of per-biosample summits
core_end	1782710	End position of core-region containing 95% of per-biosample summits
component	Digestive	Main DHS Vocabulary component

Figure 3.23: Meuleman DHS Components file structure¹³

We create tissue-specific datasets by splitting DHSs into sixteen files based on their belonging *component*. No further processing or filtering is conducted on these datasets.

3.4 Coverage Signal Normalisation Methods

Coverage plots throughout this thesis reveal differences between biosamples based on a particular DHS dataset. Therefore, all biosamples carry their biological differences, resulting in displacements of biosample coverage curves on the y-axis. Some biosamples, by nature, have a higher mean genome coverage value, whereas others have a lower mean genome coverage value. In order to abstract these biological differences from biosamples and bring coverage curves on the same scale, we perform a normalisation method:

- Surrounding Regions Median (SRM)

¹³https://zenodo.org/record/3838751/files/DHS_Index_and_Vocabulary_legend.txt?download=1

3.4.1 Surrounding Regions Median

Surrounding regions (SRs) represent regions beyond the region of interest (ROI). We describe ROI as the observed area around any genomic position (e.g. DHS peak). The reasoning behind surrounding regions is to pick genomic regions, independent of ROI and its influence, reflecting average coverage of biosample's genome.

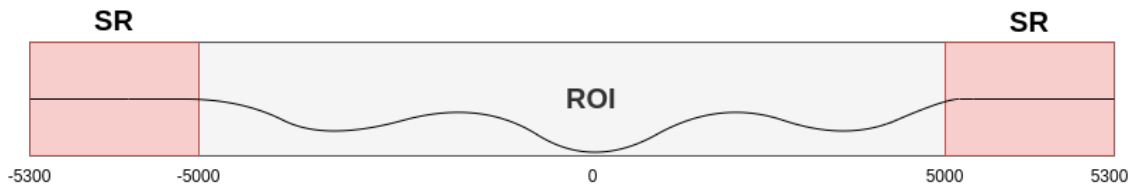


Figure 3.24: Surrounding regions schema. Black curve represents coverage signal of one biosample which is normalised by median value of two *surrounding regions* (red)

Following the theory of going beyond ROI, we need to ensure we are not running into the ROI of neighbouring DHSs. Therefore, we analyse relative distances between neighbouring peaks. In order to obtain distance information of DHSs to each other, we perform DHS dataset-based analysis. First, we take the peak of every DHS and compute its distance to the neighbouring peak. In case there is no explicit peak in the dataset defined, we take the midpoint of DHS. Likewise, we repeat the procedure for all previously generated DHS datasets. An average of those distances is shown in Figure 3.25.

3 Methods

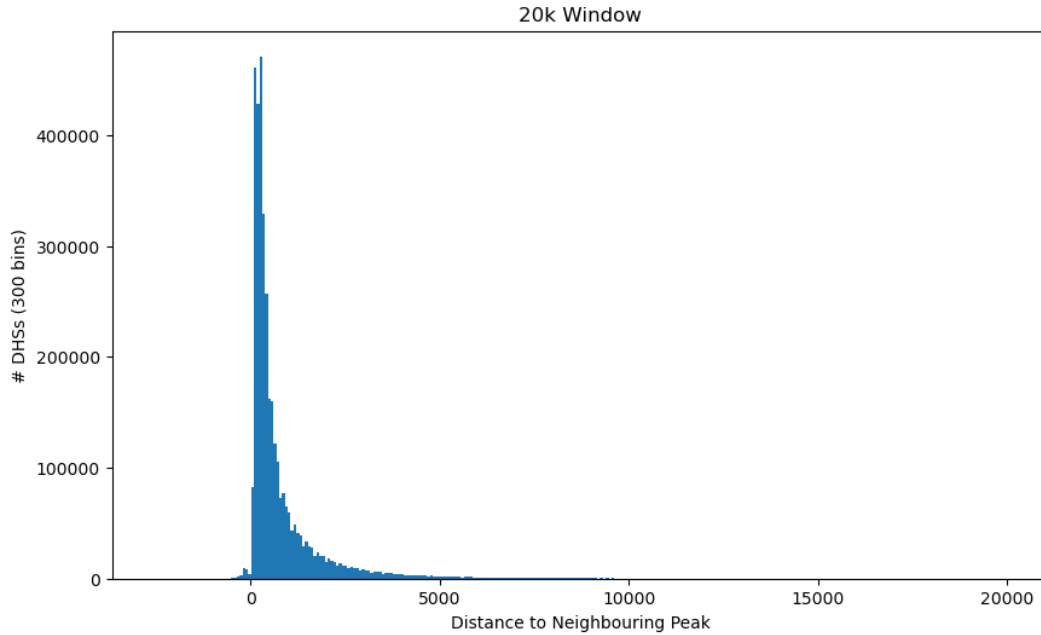


Figure 3.25: DHS distance to neighbouring peak in 20k window

Following statistics were acquired:

- 99.95% DHSs in 20k window
- 98.36% DHSs in 5k window

Based on these statistics, we opt for 10k ROI (+/- 5k) and take *surrounding regions* beyond that interval. Meaning 98.36% are already inside the ROI, and we could encounter a negligible bias from the rest 1.64% DHS, which averages out. ROI of 10k represents a good trade-off between computational speed and data confidence, as the coverage extraction script execution time tends to increase exponentially with the size of ROI.

We build coverage scripts to accept an optional parameter defining the length of *surrounding regions* whose medians are calculated and used for the coverage curve normalisation. We opt for 300bp *surrounding regions* both upstream (left) and downstream (right) in our analyses.

3.5 Multi-class Classification

To evaluate previously generated biosample and DHS datasets and corresponding coverage signals, we utilise for the classification task an ensemble learning method called *Random Forests* [Breiman, 2001]. The goal is to distinguish between healthy and diseased patients. Moreover, to distinguish between CRCs and PCs. Therefore, we perform multi-class classification.

We create a separate model for every DHS dataset because we want to evaluate the performance of individual DHS datasets. We generate classification reports containing *precision*, *recall* and *F1-score* for every model. In addition, we keep track of misclassifications and refer to the confusion matrix.

3.5.1 Feature Extraction

Since *Random Forests* is a supervised learning method, it requires non-redundant descriptive input data appropriately labelled. In order to provide the input data, we perform a feature extraction process on the previously generated normalised coverage signals (Section 3.2 and Section 3.3).

For this purpose, we utilise a Python package called **tsfresh**. This package enables the automatic calculation of a large number of time series features, and it also contains statistical tests to filter out features with low importance and explanatory power. However, the main requirement for the package is to work with time series data. Therefore, we interpret **coverage** (y-axis on coverage plots) as the **dependent variable**. Likewise, we interpret **position** (x-axis on coverage plots), captured in constant intervals (one base), as the **time**.

We perform feature extraction of biosample coverage signals using the `tsfresh extract_features()` function resulting in 1024 extracted features for every DHS dataset. Then, we filter out features with low importance and explanatory power using the `tsfresh select_features()` function supplying additional argument *multiclass = True*. This reduces the number of features between 70% - 78% depending on the DHS dataset and extracted coverage signals.

3.5.2 Model Training Method

We work with a limited set of biosamples, so we decide to perform the k -fold cross-validation method. The model's general performance will be evaluated over k splits. Realising the presence of class imbalance, we know that certain classes could get omitted entirely during the training phase (depending on the split). This leads to the model's inability to predict these classes, implying decreased accuracy.

Therefore, we opt for the **stratified** version of this method. Furthermore, we specify the argument $k = 5$ to go along with the famous 80% 20% split strategy, which means that in each iteration, 80% biosamples are used for training and 20% for testing. We perform manual hyperparameter tuning by combining the following values:

- *max_depth* – [10, 50, 100, None]
- *max_features* – ['auto', 'sqrt']
- *min_samples_leaf* – [1, 2, 5]
- *min_samples_split* – [2, 5, 10]
- *n_estimators* – [100, 200, 500, 1000, 1500, 2000]

Then, we observe the performance compared to the base RF model (default parameters).

4 Results and Discussion

4.1 Coverage Signal Normalisation Method

To compare coverage signals of different biosamples based on a DHS dataset, we performed the SRM normalisation method. Here we demonstrate the impact of this normalisation method on the coverage signal.

For demonstration purposes, we pick two biosamples (*NPH_001* and *C123_4*) from two different biosample cohorts (*HC* and *CRC*) and plot respective unnormalised coverage signals in a 10k ROI for the DHS dataset *hematopoietic_270*, as shown in Figure 4.1. We observe that the curves are displaced on the y-axis (having respective median coverage of 39,269 and 29,155). Displacements happen accordingly to the biological differences of humans. Hence, a comparison between unnormalised coverage signals would deliver incorrect results.

4 Results and Discussion

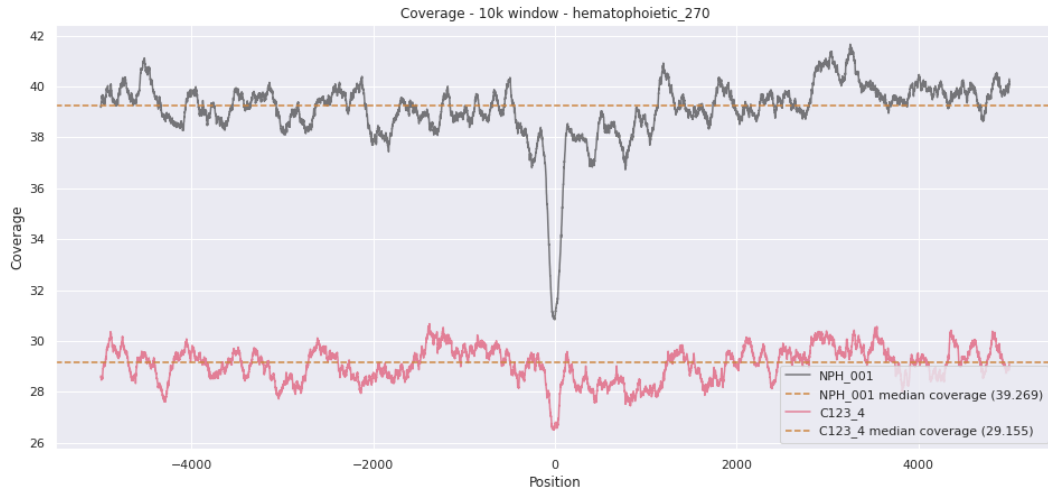


Figure 4.1: Coverage plot of two biosamples (*NPH_001* and *C123_4*) with raw coverage values for the DHS dataset *hematopoietic_270*

We perform the SRM normalisation method on the same two coverage signals from the previous plot. We observe that the curves are next to each other on the y-axis (having respective median coverage of 0,989 and 0,996). Hence, comparing these two normalised coverage signals delivers valid results.

4 Results and Discussion

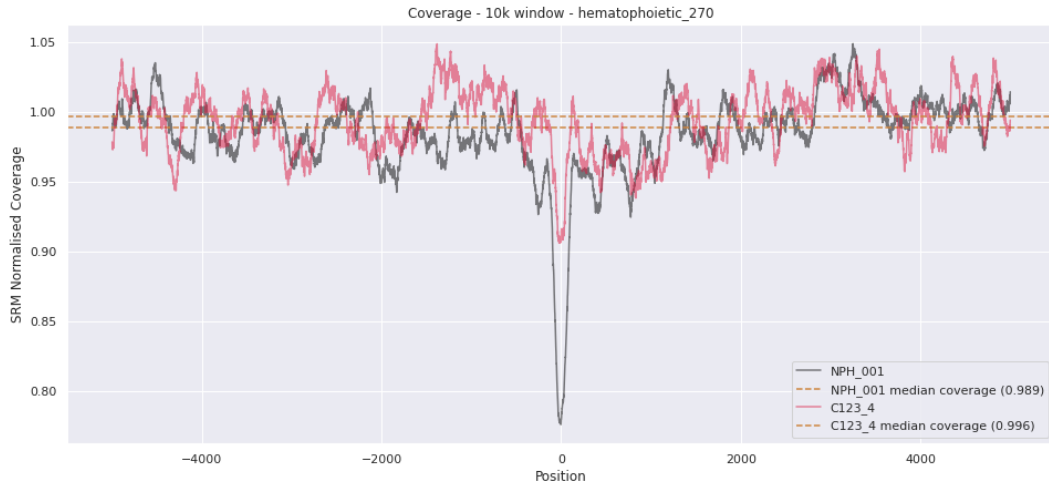


Figure 4.2: Coverage plot of two biosamples (*NPH_001* and *C123.4*) with SRM normalised coverage values for the DHS dataset *hematopoietic_270*

4.2 Coverage Plot Structure

For an easier understanding, we explain the ground structure of coverage plots and define conventions that permeate throughout plots in upcoming sections.

Figure 4.3 depicts an example of a coverage plot with three marked parts:

1. Plot title
2. Normalisation method
3. Biosample cohorts and colour scheme

Plot Title

The plot title is in accordance with the following naming convention:

{method performed}-{source of data}-{target organ/cell line}-{number of DHSs contained}

4 Results and Discussion

1. **Method performed** stands for one of the three techniques how the DHS datasets were prepared:
 - *clustering* stands for DNase-seq clustering followed by either *custom* or *bedtools* method
 - *prep* stands for statistically preprocessed
 - argument completely omitted if no processing was required
2. **Source of data** stands for one of the three sources presented in this work:
 - *Meuleman* publication [Meuleman et al., 2020]
 - *Vierstra* publication [Vierstra et al., 2020]
 - *Duke* database [Sheffield et al., 2013]
3. **Target organ/cell line**
4. **Number of DHSs contained** inside dataset

Normalisation Method

We prepend the normalisation method's name to the *Coverage* label on the y-axis. In our case, it is the **SRM** normalisation.

Biosample Groups and Color Scheme

In the coverage plots, we put focus on assessing three main biosample cohorts:

- Colorectal cancer (CRC) biosamples (coloured in *red*)
- Healthy controls (HC) (coloured in *grey* and represent reference point to all diseased biosamples)
- Prostate cancer (PC) biosamples (coloured in *blue*)

Additionally, we emphasised the *black* curve for more straightforward navigation. It represents the mean of HCs.

4 Results and Discussion

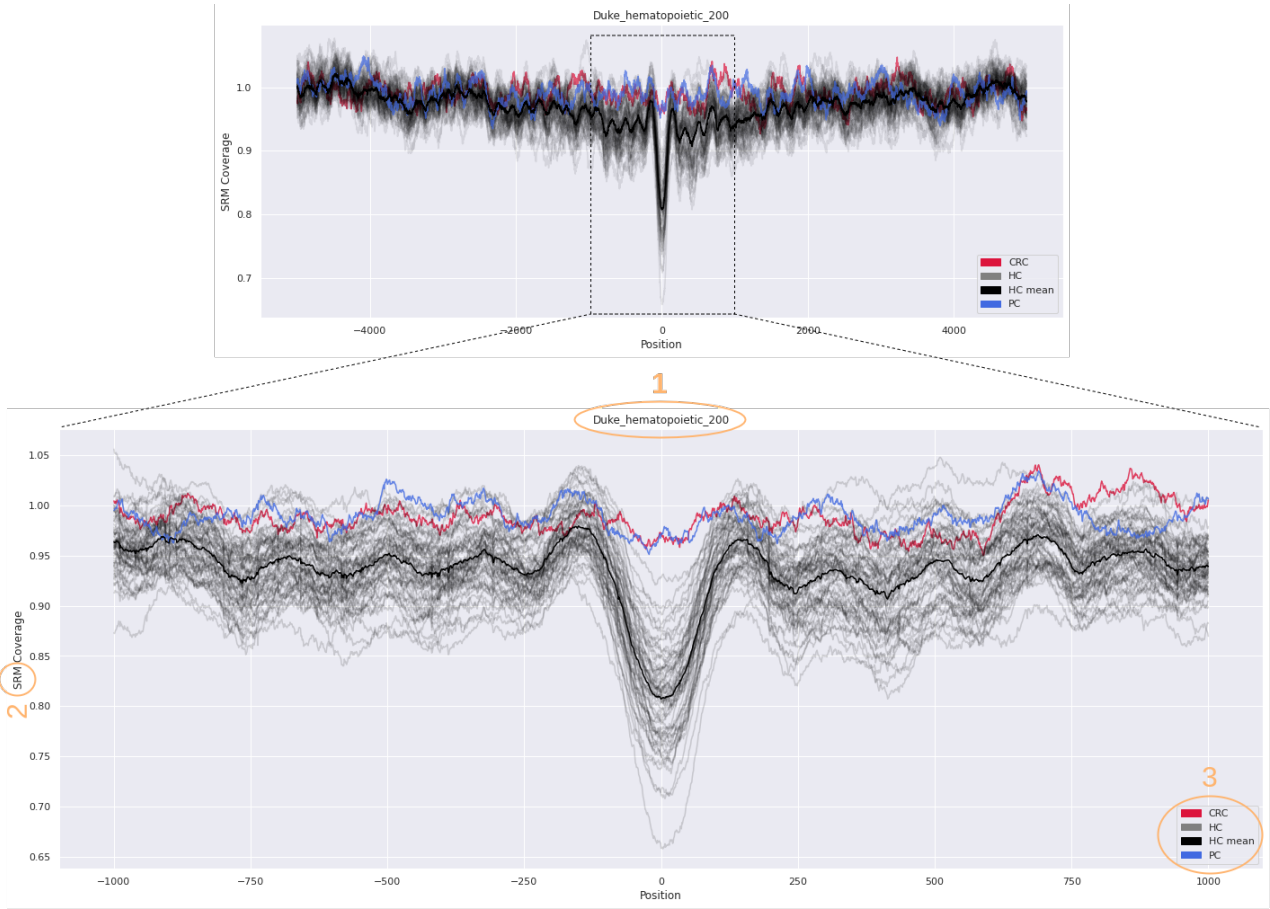


Figure 4.3: Two coverage plot examples. Coverage plot of 10k ROI window (top plot). Coverage plot of 2k ROI window with three labeled parts (bottom plot).

A signal is most unstable around the DHS peak, and we usually observe the most interesting behaviour in the 2k window around the peak. Therefore, most upcoming coverage plots are based on a 2k ROI window. Nevertheless, we would like to emphasise that we kept monitoring the 10k ROI window for any interesting signal behaviour.

4.3 Evaluation of DHS Datasets

Following evaluation is based on visual inspection of coverage plots and produced machine learning classification statistics. Even though we evaluated all the generated datasets, not all of them are present in the following. The reason for that is that we emphasised showing representative datasets for every tissue/organ targeted during dataset generation. Furthermore, while picking representative datasets, we consider different generation methods and performances of datasets to make the demonstration more versatile.

The targeted tissues/organs are:

- Hematopoietic tissue
- Prostate tissue and prostate cancer cell lines
- Digestive system (liver, colon)

4.3.1 Evaluation of Hematopoietic-based DHS Datasets

The hematopoietic tissues give rise to and house erythrocytes (red blood cells), leukocytes (white blood cells), and platelets. In addition, the hematopoietic tissues arise from hematopoietic stem cells (HSCs), including bone marrow, peripheral blood, and certain lymphoid tissue [*Hematopoietic Tissue - an overview | ScienceDirect Topics 2022*]. Therefore, we expect hematopoietic DHSs to exhibit coverage drop at DHS peak (position 0) for HCs and a flat-like signal for diseased biosamples.

Duke_hematopoietic_200

This Duke cluster represents hematopoietic tissue and contains 200 DHSs. We can observe a drop of coverage at DHS peak for all biosamples. Mainly HCs exhibit a distinct drop compared to the other two cohorts. Nevertheless, we see that certain CRC biosamples overlap with HCs.

4 Results and Discussion

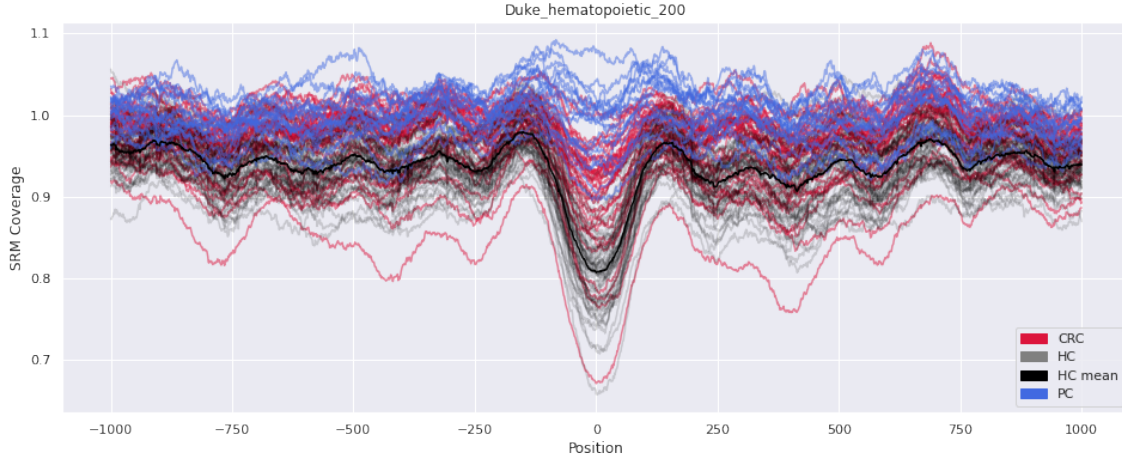


Figure 4.4: Coverage plot of Duke_hematopoietic_200 DHS dataset in 2k window

The classification report indicates an F1 score of 0.93 for HCs, which are this dataset's primary target. Therefore, we evaluate this dataset as a high-performance one. Furthermore, the confusion matrix indicates that only two HCs were misclassified, whereas 6 CRCs were misclassified as HCs (complies to what we see in Figure 4.4).

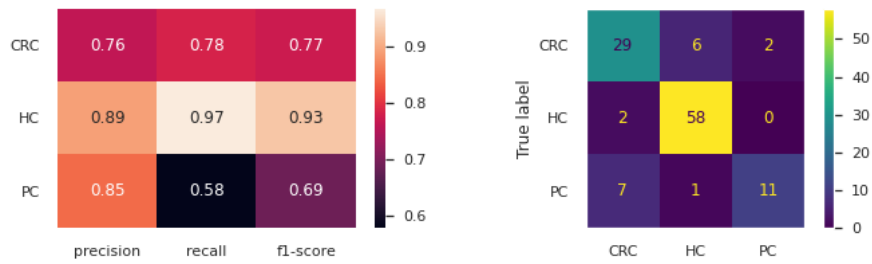


Figure 4.5: Duke_hematopoietic_200 model classification report and confusion matrix

4 Results and Discussion

Duke_hematopoietic_270

This Duke cluster represents hematopoietic tissue and contains 270 DHSs. We can observe a drop of coverage at DHS peak for all biosamples. Mainly HCs exhibit a distinct drop compared to the other two cohorts. Nevertheless, we see that certain CRC biosamples overlap with HCs.

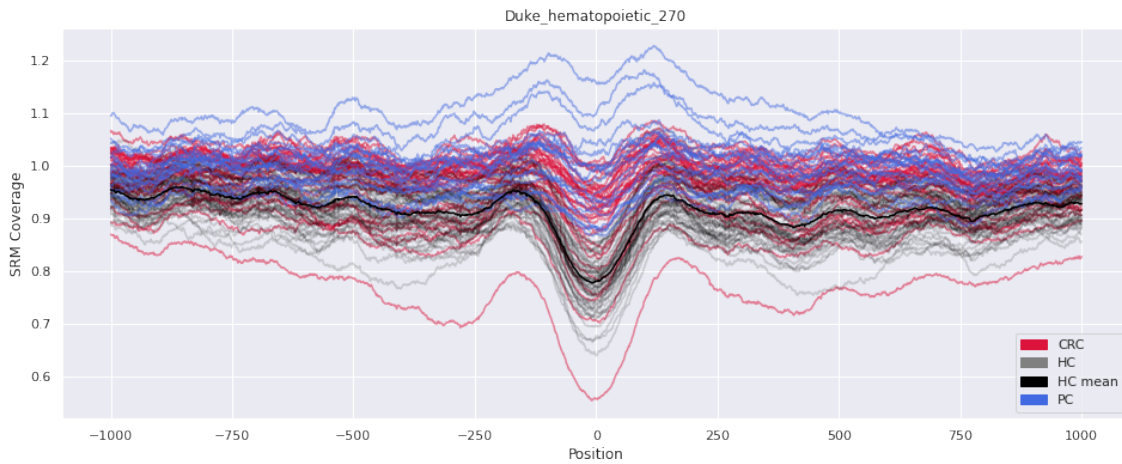


Figure 4.6: Coverage plot of Duke_hematopoietic_270 DHS dataset in 2k window

The classification report indicates an F1 score of 0.94 for HCs, which are this dataset's primary target. Therefore, we evaluate this dataset as a high-performance one, and it even performs a tiny bit better than the previous dataset. Furthermore, the confusion matrix indicates that only one HC was misclassified, whereas 6 CRCs were misclassified as HCs (complies to what we see in Figure 4.6).

4 Results and Discussion

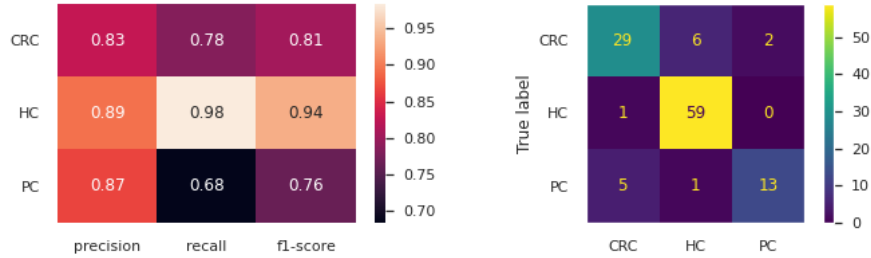


Figure 4.7: Duke.hematopoietic.270 model classification report and confusion matrix

4.3.2 Evaluation of Prostate and Prostate Cancer-based DHS Datasets

The primary targets of these DHS datasets are prostate tissue and specific prostate cancer cell lines like LNCaP and androgen-dependent LNCaP. Therefore, we expect prostate and prostate cancer-specific DHSs to exhibit coverage drop at DHS peak for PCs and a flat-like signal for HCs.

Duke_prostate_664

This Duke cluster represents the prostate tissue and contains 664 DHSs. We would expect a drop of coverage at DHS peak for PCs, but instead, the drop is present for CRCs. A few PCs exhibit an increase at the DHS peak, but the majority overlaps with HCs. There is no biological explanation for the CRC drop, making this DHS dataset unreliable.

4 Results and Discussion

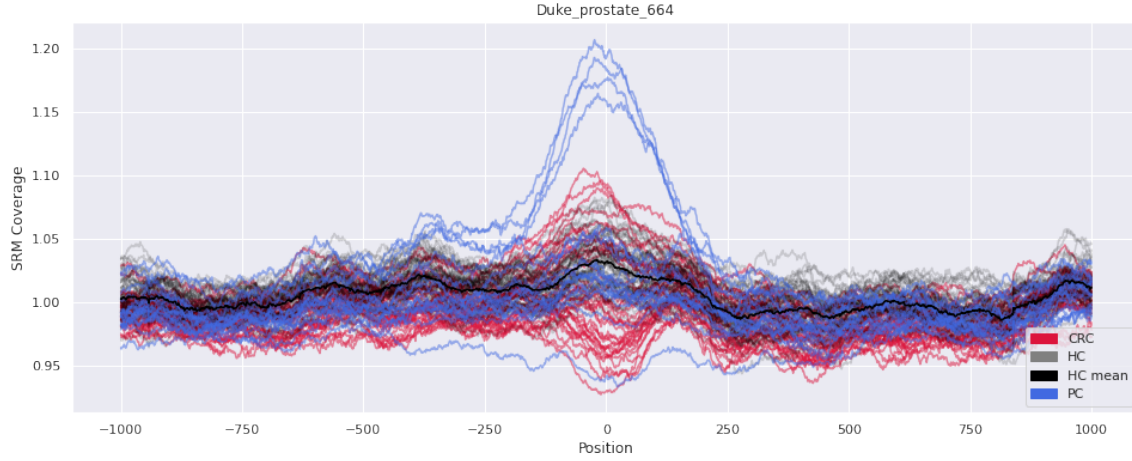


Figure 4.8: Coverage plot of Duke_hematopoietic_200 DHS dataset in 2k window

The classification report indicates an F1 score of 0.54 for PCs, which are this dataset's primary target. Therefore, we evaluate this dataset as a low-performance one. Furthermore, the confusion matrix indicates that almost two-thirds of PCs were misclassified.

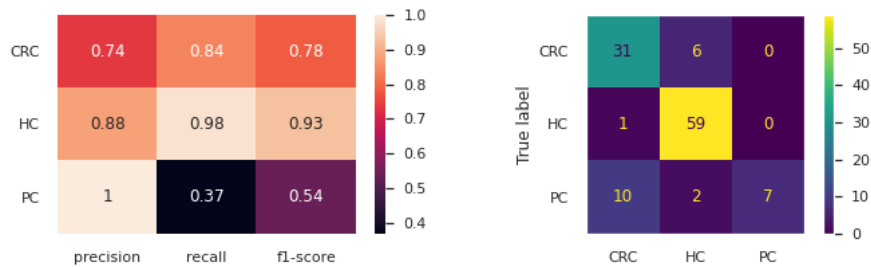


Figure 4.9: Duke_prostate_664 model classification report and confusion matrix

Duke_prostate_cancer_1135

This Duke cluster represents prostate cancer cell lines (LNCaP and androgen-dependent LNCaP) and contains 1135 DHSs. We can observe a distinct drop

4 Results and Discussion

of coverage at DHS peak for PCs. The other two cohorts overlap.

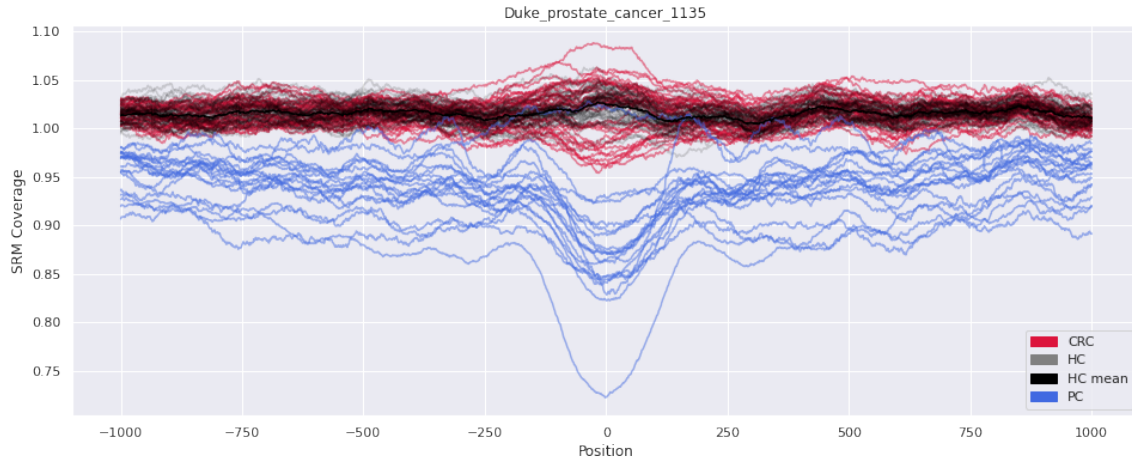


Figure 4.10: Coverage plot of Duke_prostate_cancer_1135 DHS dataset in 2k window

The classification report indicates an F1 score of 0.97 for PCs, which are this dataset's primary target. Therefore, we evaluate this dataset as a high-performance one. Furthermore, the confusion matrix indicates that only one PC was misclassified as CRC.

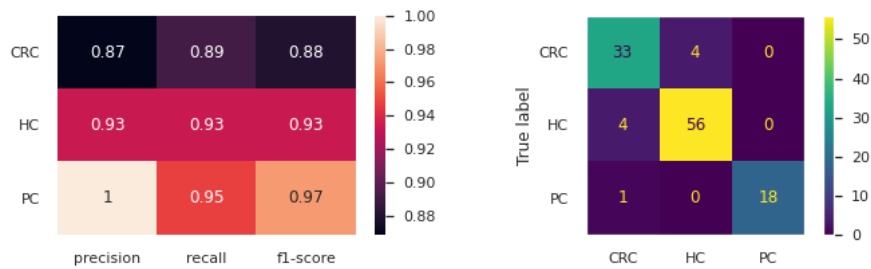


Figure 4.11: Duke_prostate_cancer_1135 model classification report and confusion matrix

4 Results and Discussion

Duke_prostate_cancer_LNCaP_andro_15236

This Duke cluster represents a prostate cancer cell line (androgen-dependent LNCaP) and contains 15236 DHSs. We can observe a distinct drop of coverage at DHS peak for PCs and a soft drop for a few CRCs between $-250bp$ and $250bp$. The rest of the HCs and CRCs overlap in the whole signal range.

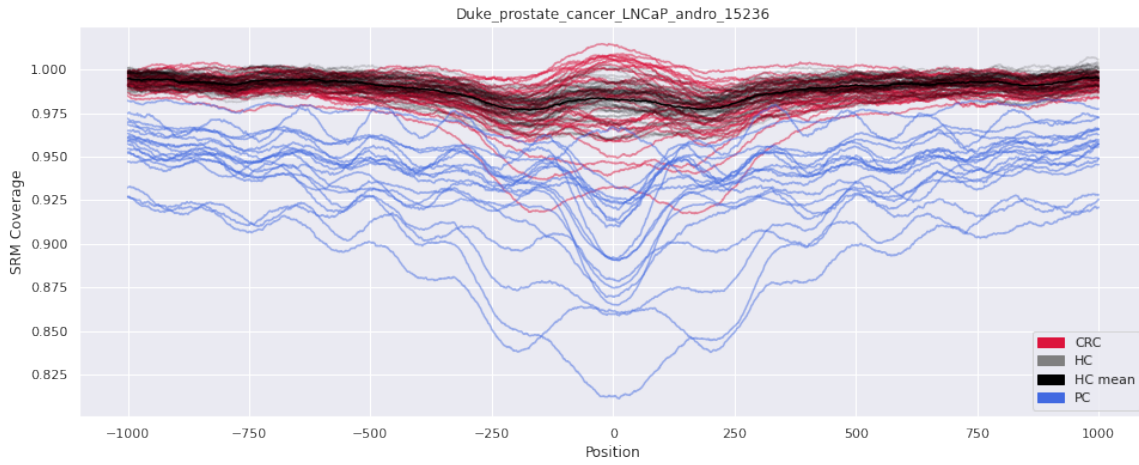


Figure 4.12: Coverage plot of Duke_prostate_cancer.LNCaP.andro.15236 in 2k window

The classification report indicates an F1 score of 0.95 for PCs, which are this dataset's primary target. Therefore, we evaluate this dataset as a high-performance one. Furthermore, the confusion matrix indicates that only one PC was misclassified as CRC.

4 Results and Discussion

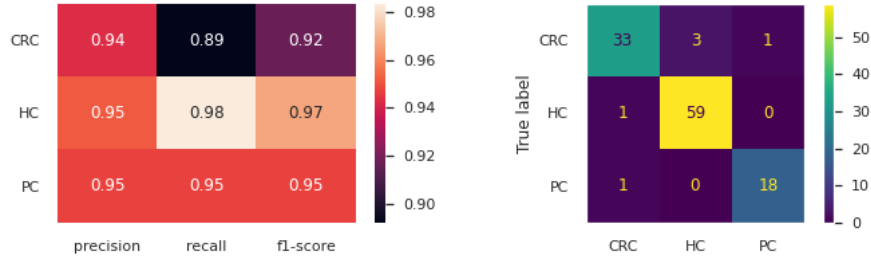


Figure 4.13: Duke_prostate_cancer_LNCaP_andro_15236 model classification report and confusion matrix

4.3.3 Evaluation of Digestive-based DHS Datasets

Two organs we target with digestive-based DHS datasets are the liver and colon. They both are part of the digestive system, which happens to be compromised in CRCs. In addition, previous studies showed that the liver contributes a distinct amount of cfDNA [Heitzer, Haque, et al., 2019, Huang et al., 2016, Lehmann-Werman et al., 2018, P. Jiang, Chan, and Lo, 2019]. Therefore, we expect liver and colon DHSs to exhibit coverage drop at the DHS peak for CRCs and a flat-like signal for HCs.

Duke_liver_437

This Duke cluster represents the liver tissue and contains 437 DHSs. We can mainly observe a coverage drop at the DHS peak for CRCs. However, we can see that the other two cohorts overlap with certain HCs.

4 Results and Discussion

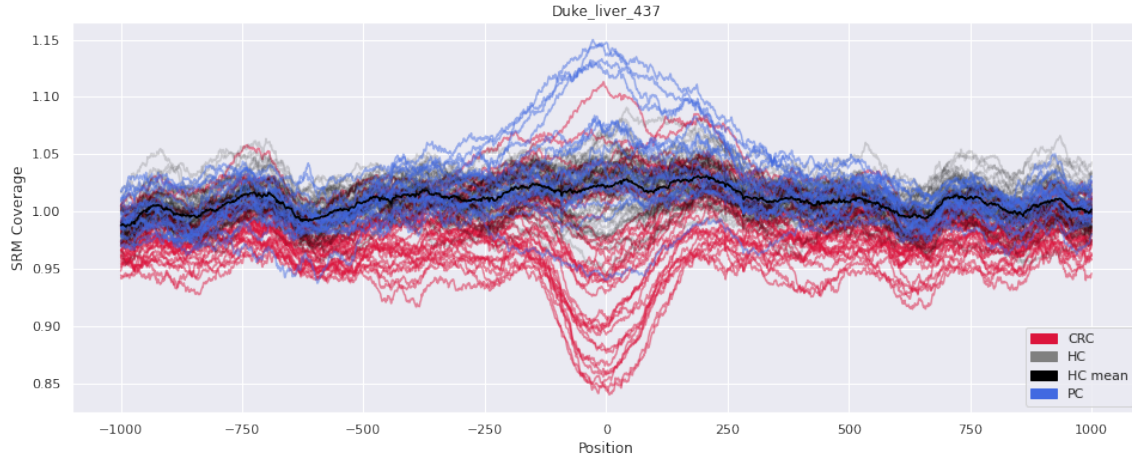


Figure 4.14: Coverage plot of Duke_liver_437 DHS dataset in 2k window

The classification report indicates an F1 score of 0.85 for CRCs, which are this dataset's primary target. Therefore, we evaluate this dataset as a mid-performance one. Furthermore, the confusion matrix indicates that six CRCs were misclassified as HCs.

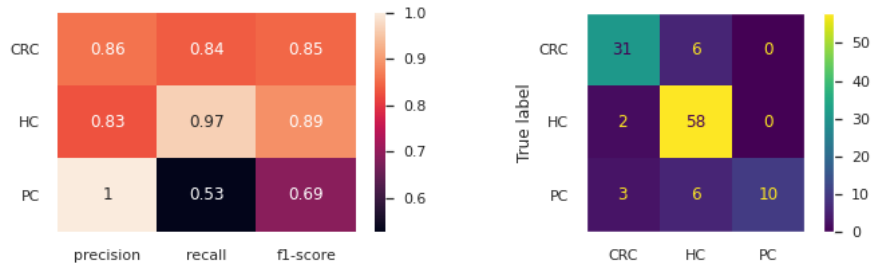


Figure 4.15: Duke.liver_437 model classification report and confusion matrix

clustering_bedtools_Vierstra_digestive_small_70p_62835

This DHS dataset represents the digestive system and contains 62835 DHSs. We can mainly observe a coverage drop at the DHS peak for CRCs, whilst

4 Results and Discussion

PCs exhibit flat and unstable signals, overlapping the other two cohorts. HCs tend to exhibit flat signals.

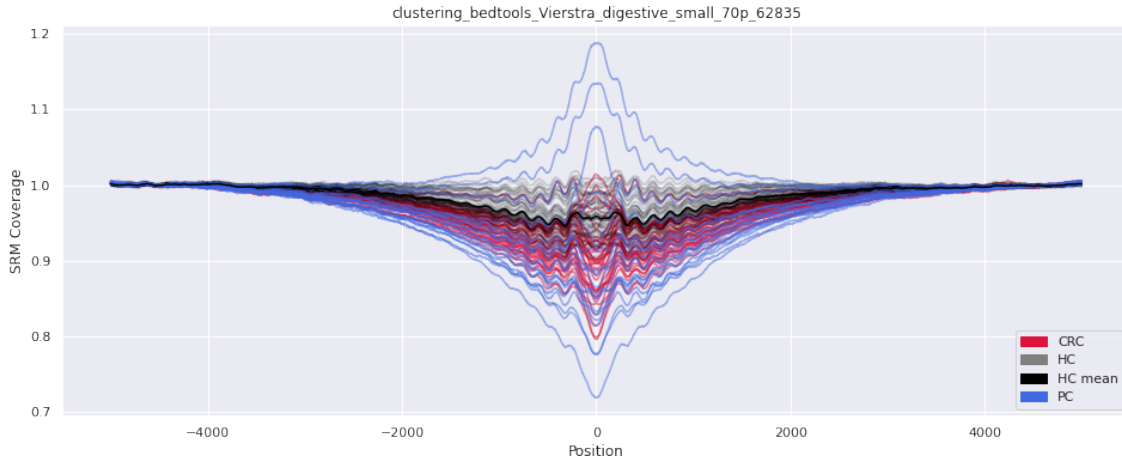


Figure 4.16: Coverage plot of clustering_bedtools.Vierstra_digestive_small_70p_62835 DHS dataset in 10k window

If we remove the PC signals because of their demonstrated behaviour, we can observe a nice separation between HCs and CRCs (whereas CRCs exhibit more distinct drop). Nevertheless, we see that certain CRC biosamples overlap with HCs.

4 Results and Discussion

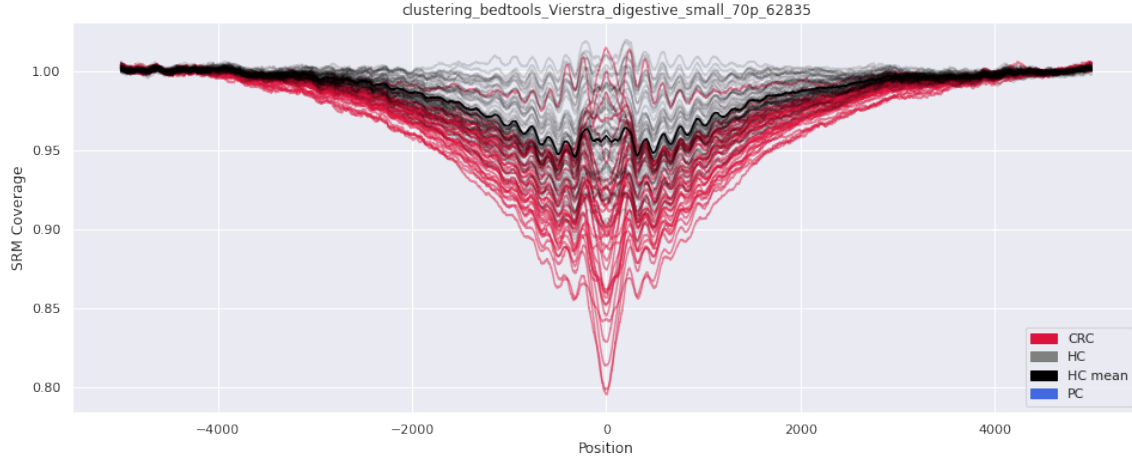


Figure 4.17: Coverage plot of clustering_bedtools_Vierstra_digestive_small_70p_62835 DHS dataset in 10k window without PCs

The classification report indicates an F1 score of 0.8 for CRCs, which are this dataset's primary target. Therefore, we evaluate this dataset as a mid-performance one. Furthermore, the confusion matrix indicates that two CRCs were misclassified as HCs and two as PCs.

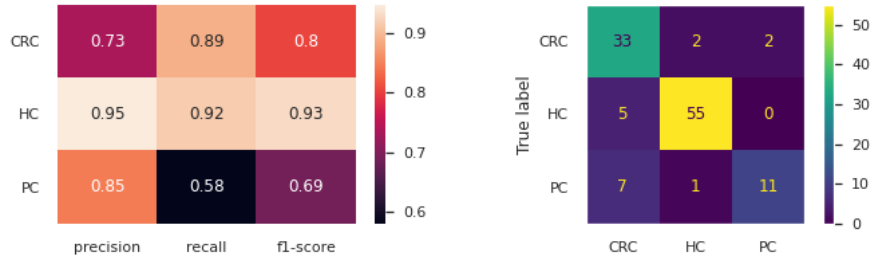


Figure 4.18: clustering_bedtools_Vierstra_digestive_small_70p_62835 model classification report and confusion matrix

4 Results and Discussion

clustering_custom_Vierstra_digestive_small_70p_52420

This DHS dataset represents the digestive system and contains 52420 DHSs. We can observe a coverage drop at the DHS peak for all biosamples. This is an improvement to the previous dataset as it corresponds to the expected behaviour. PCs tend not to show any grouping among themselves and overlap the other two biosample cohorts.

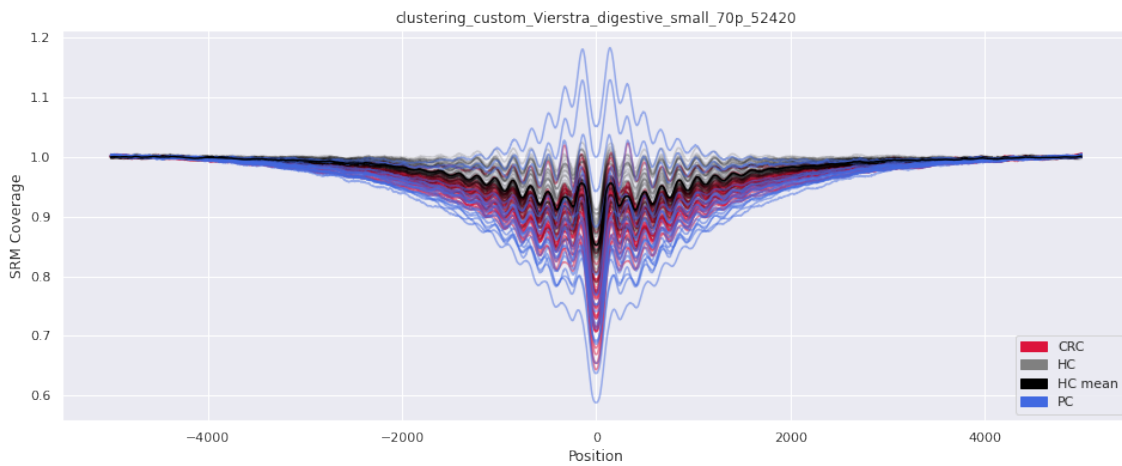


Figure 4.19: Coverage plot of clustering_custom_Vierstra_digestive_small_70p_52420 DHS dataset in 10k window

If we remove the PC signals because of their demonstrated behaviour, we can observe a nice separation between HCs and CRCs (whereas CRCs exhibit more distinct drop). Nevertheless, we see that certain CRC biosamples overlap with HCs.

4 Results and Discussion

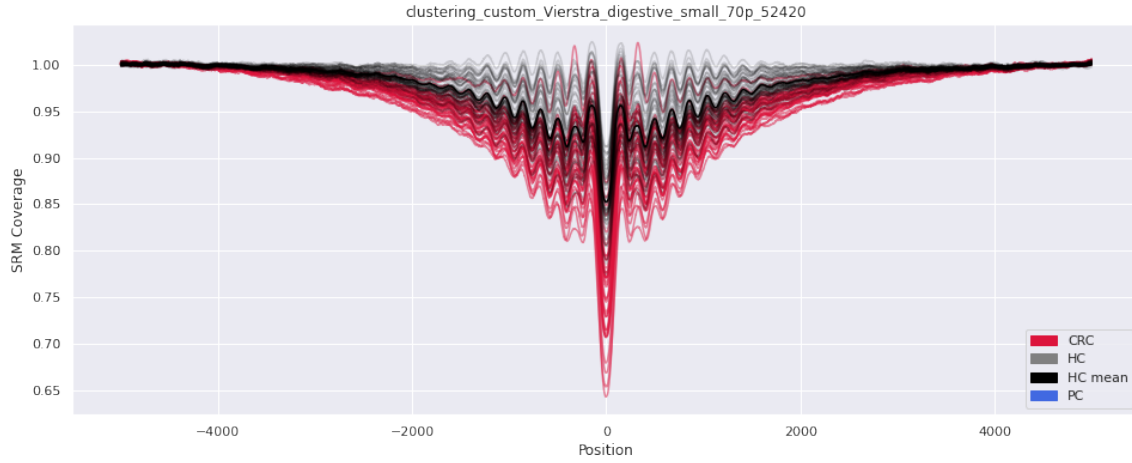


Figure 4.20: Coverage plot of clustering_custom_Vierstra_digestive_small_70p_52420 DHS dataset in 10k window without PCs

The classification report indicates an F1 score of 0.85 for CRCs, which are this dataset's primary target. Therefore, we evaluate this dataset as a mid-performance one. Furthermore, the confusion matrix indicates that two CRCs were misclassified as HCs and three as PCs.

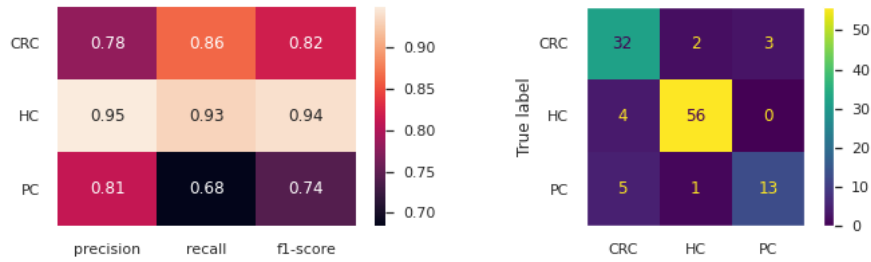


Figure 4.21: clustering_custom_Vierstra_digestive_small_70p_52420 model classification report and confusion matrix

4 Results and Discussion

prep_Meuleman_colon_win_02_5p_05_top_1k

This DHS dataset represents the digestive system and contains 62835 DHSs. We can observe a drop of coverage for CRCs, but not precisely at the DHS peak. The coverage drop happens at ~235bp downstream. We can also see that the other two biosample cohorts overlap with certain CRCs.

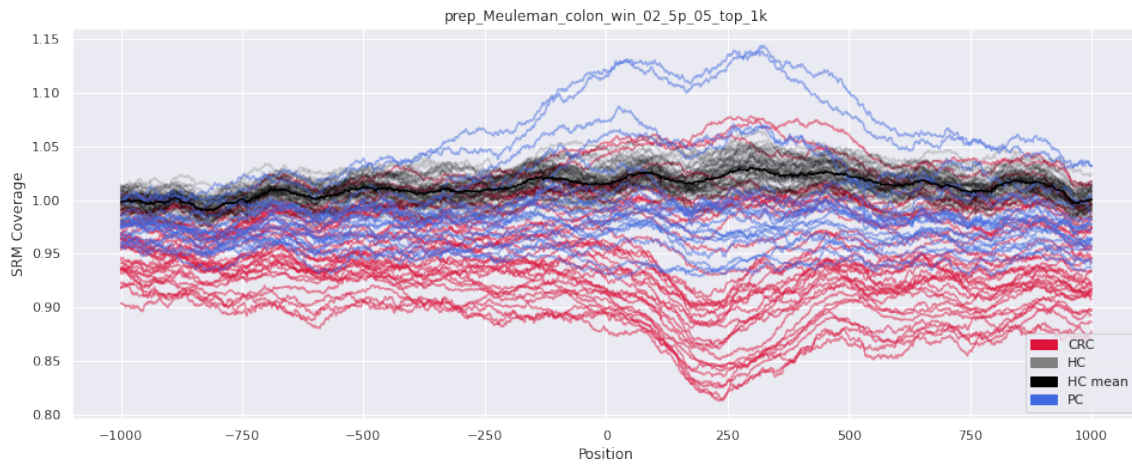


Figure 4.22: Coverage plot of prep_Meuleman_colon_win_02_5p_05_top_1k DHS dataset in 2k window

In order to get a better overview of the signal, we expand the window size to 10k. Now we can observe a more distinct coverage drop for CRCs.

4 Results and Discussion

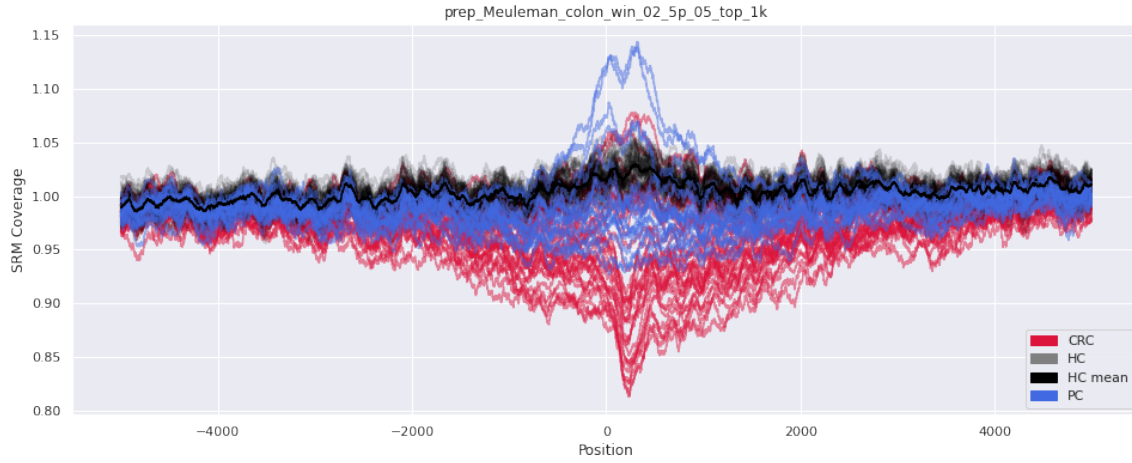


Figure 4.23: Coverage plot of prep_Meuleman_colon_win_02_5p_05_top_1k DHS dataset in 10k window

The classification report indicates an F1 score of 0.86 for CRCs, which are this dataset's primary target. Therefore, we evaluate this dataset as a mid-performance one. Furthermore, the confusion matrix indicates that three CRCs were misclassified as HCs.

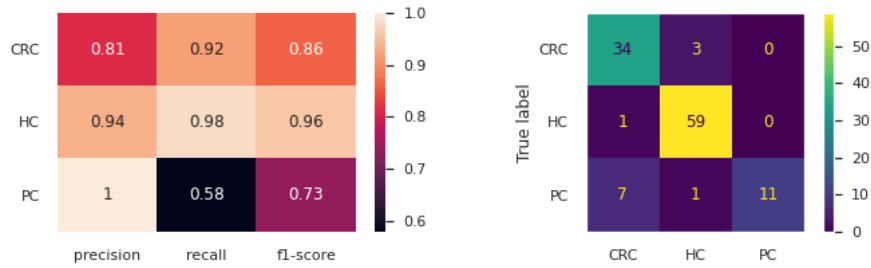


Figure 4.24: prep_Meuleman_colon_win_02_5p_05_top_1k model classification report and confusion matrix

4.4 Impact of Biosample Tumor Fraction (TF) on Coverage Signal

We recall DHS datasets whose coverage plots showed expected behaviour for most targeted biosamples, but where a small group of target biosamples still deviates from that behaviour. The model classification stats comply with deviation from expected behaviour and indicate that a small group of target biosamples were misclassified. Good examples are Figures 4.4, 4.6, 4.12, 4.17 and 4.20.

As an example, we take the Duke_hematopoietic_200 DHS dataset and investigate CRC suspects overlapping HCs. We identified these CRCs (C219_6, C127_10, C219_8, C139_10, C140_11, C218_4, C109_8, C208_10, C219_6) and consecutively performed tumor fraction estimation method [Adalsteinsson et al., 2017]. Respective tumor fraction values are between 0% - 16%. Therefore, we acknowledge an impactful classification limiting factor for some DHS datasets.

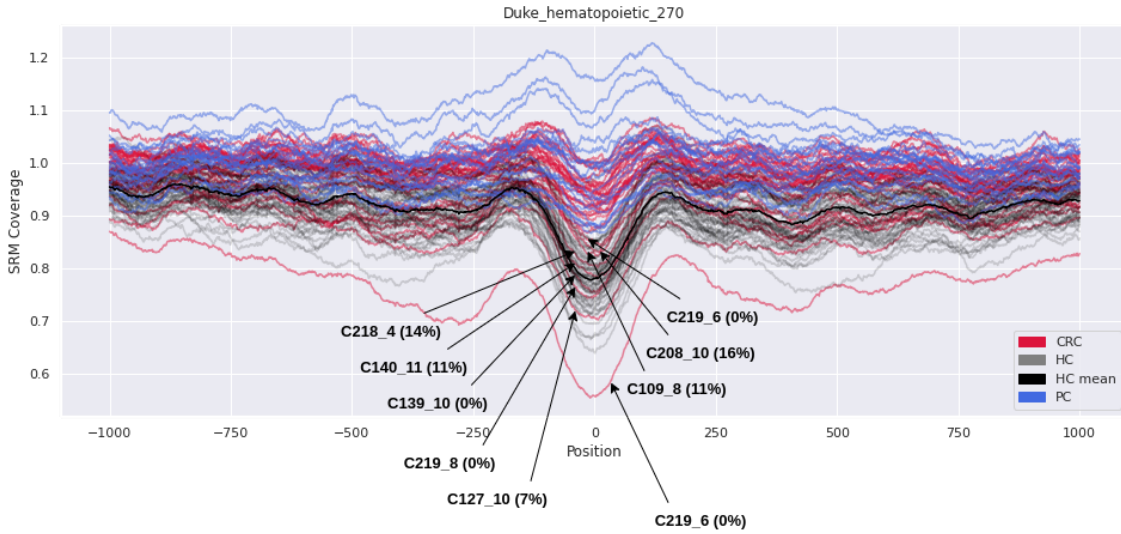


Figure 4.25: Coverage plot of Duke_hematopoietic_270 DHS dataset in 2k window with overlapping CRCs and their tumor fraction labels

4 Results and Discussion

Additionally, we generated a misclassification bar chart to confirm the suspects. The misclassification bar chart complies indeed with the previously listed CRCs.

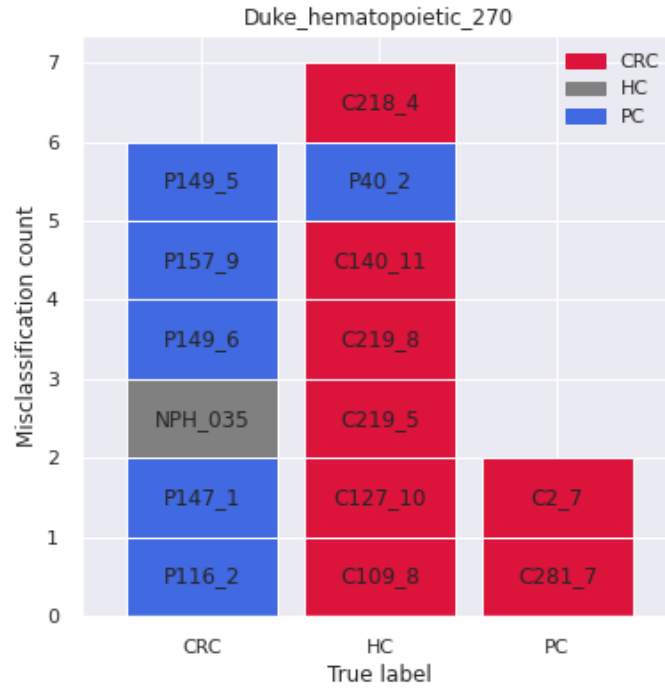


Figure 4.26: Misclassification bar chart of Duke.hematopoietic_270 model

5 Conclusion

This thesis challenged unconventional diagnostic methods in medicine, combining them with computer science. We aimed to employ machine learning methods and specific genomic regions, the DHSs, to distinguish between healthy and cancer patients moreover between different cohorts of cancer patients.

First, we showed what it takes to prepare the biosamples for upcoming analyses. Then, we performed two approaches to generate representative DHSs targeting specific cancer types. The first approach involves clustering DNase-seq data, whereas the second approach involves pre-clustered DHSs. To handle the clustering of DNase-seq data, we developed two methods: custom clustering and bedtools-based clustering. For the pre-clustered DHSs, we performed domain expert validation and statistical preprocessing. Consecutively, to evaluate the performance of generated DHS datasets, we created coverage extraction scripts delivering respective biosample coverage signals.

Based on the quantitative and qualitative analysis of DHS datasets and biosample coverage signals in response to different cancer types, it can be concluded that the accessibility patterns of individual DHS datasets play a critical role. We evaluated multiple DHS datasets per cancer type, and the results indicate that most of them exhibit expected behaviour. However, we also showed cases where the DHS dataset failed to exhibit expected behaviour for the targeted biosample cohort, resulting in an unsuccessful classification. Being able to perform classification with the majority of generated DHS datasets, we showed that DHSs are indeed related to cancer biology.

5.1 Future Work

During the research, specific topics struck out as exciting but are out of the thesis' scope. Hence, we would like to address them here as future work.

We pointed out the TF as a limiting factor in the coverage signal distinction and classification. In case it is too low, the affected biosamples become undetectable. TF as a limiting factor obstructs a use-case in early cancer detection, as the TFs in early cancer stages tend to be below the shown threshold. It could be further investigated to gain deeper insight and eventually eliminated.

DNase-seq data clustering is just a part of one of three subgoals in this thesis, which designates that the main focus was not on the refinement of clustering methods. However, the clustering methods are a deciding factor for the resulting DHS dataset. Therefore, we suspect that more sophisticated clustering methods could deliver better-performing DHS datasets than the ones we generated. Undeniably, there is also dependency on contained information and quality of DNase-seq data.

Biosample coverage signal is averaged over all DHSs belonging to a dataset. However, coverage signals of individual DHSs are relatively noisy. This noise poses a problem for individual DHS evaluation. Therefore, developing an evaluation method for individual DHS noisy coverage signals could be a game-changer in eliminating low-performance DHSs.

While the `tsfresh` package automates feature extraction based on the predefined set of features, it might be necessary to incorporate additional custom features to improve the models. One of the potentially relevant custom features could be the area over curve (AOC), which could improve the model accuracy.

High coverage sequencing is mainly used for research purposes. Moreover, the costs are too high for routine diagnostics. On the other hand, low coverage implies less data, posing a challenge to the quality of coverage signals. Hence, a future goal could be to accomplish similar performance as shown, just for low-coverage biosamples while retaining high-coverage model accuracies.

Bibliography

- Adalsteinsson, Viktor A. et al. (2017). "Scalable whole-exome sequencing of cell-free DNA reveals high concordance with metastatic tumors." In: *Nat. Commun.* 8. DOI: 10.1038/s41467-017-00965-y (cit. on p. 73).
- Amemiya, Haley M., Anshul Kundaje, and Alan P. Boyle (June 2019). "The ENCODE Blacklist: Identification of Problematic Regions of the Genome - Scientific Reports." In: *Sci. Rep.* 9.9354, pp. 1–5. ISSN: 2045-2322. DOI: 10.1038/s41598-019-45839-z.
- Andrews, Simon et al. (Jan. 2012). *FastQC*. Babraham Institute. Babraham, UK.
- Barbier, Jérémy et al. (June 2021). "Coupling between Sequence-Mediated Nucleosome Organization and Genome Evolution." In: *Genes* 12.6, p. 851. ISSN: 2073-4425. DOI: 10.3390/genes12060851.
- Bioinformatics, EcSeq (Aug. 2016). *Trimming adapter sequences - is it necessary?* [Online; accessed 14. Aug. 2021]. URL: <https://www.ecseq.com/support/ngs/trimming-adapter-sequences-is-it-necessary> (cit. on p. 23).
- Breiman, Leo (Oct. 2001). "Random Forests." In: *Machine Learning* 45.1, pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324 (cit. on pp. 8, 51).
- Christ, Maximilian et al. (May 2018). "Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)." In: *Neurocomputing* 307. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2018.03.067.
- D'Antonio, Matteo et al. (Sept. 2017). "Identifying DNase I hypersensitive sites as driver distal regulatory elements in breast cancer." In: *Nat. Commun.* 8.436, pp. 1–16. ISSN: 2041-1723. DOI: 10.1038/s41467-017-00100-x (cit. on p. 1).

Bibliography

- Degner, Jacob F. et al. (Feb. 2012). "DNase I sensitivity QTLs are a major determinant of human expression variation - Nature." In: *Nature* 482, pp. 390–394. ISSN: 1476-4687. DOI: 10.1038/nature10808 (cit. on p. 13).
- Dwivedi, Dhruva J. et al. (Aug. 2012). "Prognostic utility and characterization of cell-free DNA in patients with severe sepsis." In: *Crit. Care* 16.4, pp. 1–11. ISSN: 1364-8535. DOI: 10.1186/cc11466.
- Farman, Farman Ullah et al. (Apr. 2018). "Nucleosomes positioning around transcriptional start site of tumor suppressor (Rbl2/p130) gene in breast cancer." In: *Mol. Biol. Rep.* 45.2, pp. 185–194. ISSN: 1573-4978. DOI: 10.1007/s11033-018-4151-6. eprint: 29417345.
- Genome Browser FAQ* (Sept. 2021). [Online; accessed 26. Sep. 2021]. URL: <http://genome.cse.ucsc.edu/FAQ/FAQformat.html#format1>.
- Hahn, Andrew W. et al. (Jan. 2019). "Cell-free Circulating Tumor DNA (ctDNA) in Metastatic Renal Cell Carcinoma (mRCC): Current Knowledge and Potential Uses." In: *Kidney Cancer* 3.1, pp. 7–13. ISSN: 2468-4562. DOI: 10.3233/KCA-180048 (cit. on p. 14).
- Hartmann, G. (Jan. 2017). "Nucleic Acid Immunity." In: *Advances in Immunology*. Vol. 133. Cambridge, MA, USA: Academic Press, pp. 121–169. DOI: 10.1016/bs.ai.2016.11.001 (cit. on p. 12).
- Heitzer, Ellen, Imran S. Haque, et al. (Feb. 2019). "Current and future perspectives of liquid biopsies in genomics-driven oncology." In: *Nat. Rev. Genet.* 20.2, pp. 71–88. ISSN: 1471-0064. DOI: 10.1038/s41576-018-0071-5. eprint: 30410101 (cit. on pp. 14, 65).
- Heitzer, Ellen and Michael R. Speicher (Nov. 2018). "One size does not fit all: Size-based plasma DNA diagnostics." In: *Sci. Transl. Med.* 10.466, eaav3873. ISSN: 1946-6242. DOI: 10.1126/scitranslmed.aav3873. eprint: 30404860 (cit. on p. 9).
- Hematopoietic Tissue - an overview | ScienceDirect Topics* (Feb. 2022). [Online; accessed 26. Feb. 2022]. DOI: 10.1016/B978-0-323-53045-3.00023-4 (cit. on p. 58).
- Huang, Ao et al. (2016). "Plasma Circulating Cell-free DNA Integrity as a Promising Biomarker for Diagnosis and Surveillance in Patients with Hepatocellular Carcinoma." In: *J. Cancer* 7.13, p. 1798. DOI: 10.7150/jca.15618 (cit. on p. 65).
- Jiang, Hongshan et al. (Dec. 2014). "Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads." In: *BMC Bioinf.* 15.1, pp. 1–12. ISSN: 1471-2105. DOI: 10.1186/1471-2105-15-182.

Bibliography

- Jiang, Peiyong, K. C. Allen Chan, and Y. M. Dennis Lo (Aug. 2019). "Liver-derived cell-free nucleic acids in plasma: Biology and applications in liquid biopsies." In: *J. Hepatol.* 71.2, pp. 409–421. ISSN: 0168-8278. DOI: 10.1016/j.jhep.2019.04.003 (cit. on p. 65).
- Lehmann-Werman, Roni et al. (June 2018). "Monitoring liver damage using hepatocyte-specific methylation markers in cell-free circulating DNA." In: *JCI Insight* 3.12. ISSN: 0021-9738. DOI: 10.1172/jci.insight.120687 (cit. on p. 65).
- Ling, G. and D. Waxman (2013). "Isolation of nuclei for use in genome-wide DNase hypersensitivity assays to probe chromatin structure." In: *undefined*. URL: <https://www.semanticscholar.org/paper/Isolation-of-nuclei-for-use-in-genome-wide-DNase-to-Ling-Waxman/95490139c9be85f2f35177b1e9243498e77ec748> (cit. on p. 13).
- Meuleman, Wouter et al. (Aug. 2020). "Index and biological spectrum of human DNase I hypersensitive sites - Nature." In: *Nature* 584, pp. 244–251. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2559-3 (cit. on pp. 5, 6, 33, 56).
- Mitchelson, K. R. (Jan. 2005). "DNA SEQUENCING." In: *Encyclopedia of Analytical Science (Second Edition)*. Waltham, MA, USA: Elsevier, pp. 286–293. ISBN: 978-0-12-369397-6. DOI: 10.1016/B0-12-369397-7/00683-X (cit. on p. 10).
- Moore, Jill E. et al. (July 2020). "Expanded encyclopaedias of DNA elements in the human and mouse genomes - Nature." In: *Nature* 583, pp. 699–710. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2493-4 (cit. on p. 13).
- Neph, Shane et al. (Sept. 2012). "An expansive human regulatory lexicon encoded in transcription factor footprints - Nature." In: *Nature* 489, pp. 83–90. ISSN: 1476-4687. DOI: 10.1038/nature11212.
- O'Leary, Nuala A. et al. (Jan. 2016). "Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation." In: *Nucleic Acids Res.* 44.Database, p. D733. DOI: 10.1093/nar/gkv1189.
- Pareek, Chandra Shekhar, Rafal Smoczynski, and Andrzej Tretyn (2011). "Sequencing technologies and genome sequencing." In: *Journal of Applied Genetics* 52.4, p. 413. DOI: 10.1007/s13353-011-0057-x (cit. on p. 10).
- Pinzani, Pamela et al. (June 2021). "Updates on liquid biopsy: current trends and future perspectives for clinical application in solid tumors." In: *Clinical Chemistry and Laboratory Medicine (CCLM)* 59.7, pp. 1181–1200. ISSN: 1437-4331. DOI: 10.1515/cclm-2020-1685 (cit. on p. 10).

Bibliography

- Pradhan, Dibyabhaba et al. (Jan. 2019). "High-throughput sequencing." In: *Data Processing Handbook for Complex Biological Data Sources*. Cambridge, MA, USA: Academic Press, pp. 39–52. ISBN: 978-0-12-816548-5. DOI: 10.1016/B978-0-12-816548-5.00004-6 (cit. on p. 10).
- Sheffield, Nathan C. et al. (Mar. 2013). "Patterns of regulatory activity across diverse human cell types predict tissue identity, transcription factor binding, and long-range interactions." In: *Genome Res.* 23.5, pp. 777–788. ISSN: 1088-9051. DOI: 10.1101/gr.152140.112 (cit. on pp. 6, 33, 56).
- Song, Lingyun and Gregory E. Crawford (Feb. 2010). "DNase-seq: a high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells." In: *Cold Spring Harbor protocols* 2010.2, pdb.prot5384. DOI: 10.1101/pdb.prot5384 (cit. on p. 12).
- Sturm, Marc, Christopher Schroeder, and Peter Bauer (Dec. 2016). "SeqPurge: highly-sensitive adapter trimming for paired-end NGS data." In: *BMC Bioinf.* 17.1, pp. 1–7. ISSN: 1471-2105. DOI: 10.1186/s12859-016-1069-7.
- Ulz, Peter (Oct. 2016). *Nucleosome protection of circulating tumor DNA*. [Online; accessed 21. Sep. 2021]. URL: <https://diglib.tugraz.at/download.php?id=5891c8e483966&location=browse>.
- Ulz, Peter et al. (Oct. 2016). "Inferring expressed genes by whole-genome sequencing of plasma DNA - Nature Genetics." In: *Nat. Genet.* 48, pp. 1273–1278. ISSN: 1546-1718. DOI: 10.1038/ng.3648.
- Vierstra, Jeff et al. (July 2020). "Global reference mapping of human transcription factor footprints - Nature." In: *Nature* 583, pp. 729–736. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2528-x (cit. on pp. 4, 26, 56).
- Weintraub, Harold and Mark Groudine (Sept. 1976). "Chromosomal Subunits in Active Genes Have an Altered Conformation." In: *Science*. URL: <https://www.science.org/doi/abs/10.1126/science.948749> (cit. on p. 12).
- Winter, Deborah R. et al. (May 2013). "DNase-seq predicts regions of rotational nucleosome stability across diverse human cell types." In: *Genome Res.* 23.7, pp. 1118–1129. ISSN: 1088-9051. DOI: 10.1101/gr.150482.112 (cit. on p. 13).
- Zhong, Jianling et al. (Jan. 2016). "Mapping nucleosome positions using DNase-seq." In: *Genome Res.* 26.3, pp. 351–364. ISSN: 1088-9051. DOI: 10.1101/gr.195602.115 (cit. on p. 13).