UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science
Computer Science

Hans Peeter Tulmin

Pairwise DNA Methylation Analysis

Bachelor's thesis (9 ECTS)

Supervisor: Dr. Balaji Rajashekar

Autor: .... " ......" mai 2015

Juhendaja: .... " ......" mai 2015

Lubada kaitsmisele

Professor: .... " ......" mai 2015

Tartu 2015

Pairwise DNA Methylation Analysis

Abstract: The ever-increasing interest in genetics has lead to the discovery of epigenetic alterations - cellular trait variations not caused by the changes in the DNA sequence. DNA mehtylation is the most researched one of the alterations, being the addition of a methyl group to a DNA nucleobase. Even though methylation itself has been reseached, it is often considered on a per-read basis. The pairwise influence of methylation sites is still relatively unexplored. One of the reasons for this is that it is difficult to conduct such an experiment, mainly due to the lack of tools. The matters are worse with pairwise methylation, as only a few papers have been published about the topic. The aim of this thesis is to provide insight in conducting an analysis of pairwise DNA methylation, and to provide a tool as an R package for extracting pairwise methylation values.
Keywords: DNA methylation, pairwise methylation, R

DNA metülatsiooni analüüs paaride kaupa

Lühikokkuvõte: Üha süvenev huvi geneetikasse on kaasa toonud epigeneetiliste modifikatsioonide avastamise. Epigeneetilisteks modifikatsioonideks loetakse rakulisi tunnuseid mis pole põhjustatud DNA sekventsis esinevatest erinevustest. Sellistest modifikatsioonidest on enim uuritud DNA metülatsiooni, mis seisneb DNA nukleobaasile metüülrühma lisandumises. Vaatamata sellele, et metülatsiooni ennast on uuritud, vaadatakse metülatsiooni enamasti lugemite kaupa. See, kuidas metülatsiooni esinemisega seotud nukleobaasid teineteist mõjutavad, on suhteliselt tundmatu. Üheks põhjuseks, miks DNA paaride kaupa metülatsiooni piisavalt uuritud pole, võib pidada tööriistade vähesusest. Kahjuks on selle valdkonna kohta avaldatud vaid paar uurimustööd. Selle bakalaureusetöö eesmärgiks on anda lugejale ülevaade paaride kaupa DNA metülatsiooniga seotud eksperimendi läbiviimisest ja esitleda tööriista R paketi kujul, mis võimaldab koguda lugemitest andmeid paaride kaupa metülatsiooni kohta.
Märksõnad:DNA metülatsioon, paarikaupa metülatsioon, R

Contents

1 Introduction

By the course of evolution, organisms have changed drastically. Going from the single celled to the complex life forms we encounter today. Unfortunately, the notion of genetics in its modern form was established only at the mid 19th century. This essentially means even though we are the successful products of hundreds of generations of evolution, we still really do not know nor have had the time to study what we are or how we function.
Just recently, with the help of the revolution of computers, have we had a chance to peek into the building blocks of living organisms. This has lead to the discovery of DNA. Following the discovery, we have observed various organisms and annotated their genetical structure, We have managed to connect various diseases to certain regions of this structure and have also managed to create reference genomes for the genetical structure we expect to see in an organism. This is used in comparing individuals with specific traits to the reference and find deviations of interest.

However, this is not the only information encoded in our cells. Hidden from the observations of genetic code are epigenetic factors, of whom the most is known is DNA methylation. Currently, there are adequate tools and extensions for finding methylation in DNA reads. Methylation is usually considered considered on a per-read basis, giving each read or region an aggregated value as a methylation status.

There is a different approach extracting methylation from the reads as well. This involves mapping all the reads back to the genome to identify all methylation sites found in the reads. Then the reads can be compared against the sites, identifying methylation sites adjacent in the reads. Such a method focuses on the methylation of a certain site instead of a read. This, however, opens up more possibilities. For example, comparing the methylation of two adjacent sites to find out how they influence one-another.

The aim of this thesis is to provide insight in the analysis of pairwise methylation and to provide a method to extract methylation values from reads mapped to a fitting genome. The goal is to give a useful tool for further analysis, along with a manual on how to conduct experiments and how to interpret the results of such an analaysis.

2 Preliminaries

2.1 DNA

DNA (Deoxyribonucleic acid)[1] is a molecule used by all living organisms to encode and store genetic instructions for the development of the organism. Due to being used as a storage, DNA has to be stable - keep a fairly stable state throughout the lifespan of the cell it is contained in. For this, DNA uses a double helix structure, meaning that for each nucleobase(building block of DNA) there exists a complement base. The building blocks themselves are adenine, guanine, thymine and cytosine.

The double helix structure means that for each base(nucleobase) there exists a complementary base on the other strand. The strands themselves are wrapped around each other to preserve the contained bases better. For the complements, adenine is complement to thymine and guanine is complement to cytosine, meaning that if we see an adenine on one of the strands, we know that there will be a thymine on the other strand in the same place. The DNA structure is kept together on a strand basis by phosphate between each two nucleobases on the strand.

Another notion we need to establish before moving on is the definition of a DNA sequence. What we call a DNA sequence is essentially a string of length N, that is composed of character representations of the nucleotides(nucleobases). Thus, we have a set S = {A,C,G,T } that forms the alphabet for our strings.

The reference genome is also a DNA sequence, but with a few more elements in the alphabet set. Since a reference genome is supposed to represent an organism in general, it is required to represent randomness as well. For example, some nucleobases may be completely random across organisms of the same species. Such bases are denoted as N in the reference genome. Also, in some cases, the randomness might be restricted. For example 5-methylcytosine(the most common methylation) occasionally deaminates to thymine and ammonia. This is a pretty common mutation, but the it makes sense to note such a base as cytosine/thymine.

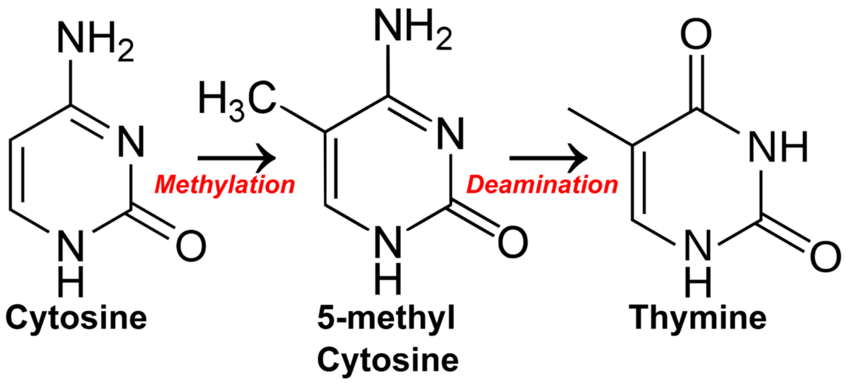

Figure 2.1: The methylation and deamination of cytosine

2.2 Methylation

In general, methylation[2] is a process of adding a methyl group to a substance, either by substitution or a reaction that causes the methyl group to occur.

DNA methylation is the addition of a methyl group to a base pair. The most common methylation is the transformation of cytosine to 5-methylcytosine. According to current knowledge, this is the only type of methylation that occurs in mammalian cells, and is of greater interest because of this property. There are other methylation types present, but they haven't been explored much. Furthermore, mammalian DNA methylation occurs mostly in CpG(Cytosine-phosphate-Guanine) context[2].

For the analysis of methylation in cytosine, it has been discovered that treating it with bisulfite causes it to deaminate into uracil[3]. This means that an amino group is removed from the cytosine molecule. However, such a reaction only occurs in cytosine molecules which are not methylated. This essentially enables us to identify methylation in cytosine and is the method used for extracting methylation. Such a technique is actually called bisulfite sequencing.

2.3 File formats and specifications

2.3.1 FASTA

FASTA [4] is a text based file format representing either nucleotide or peptide sequences. Each of the nucleotides or amino acids are represented with a single character.

Status	Abbreviation
Adenine	A
Thymine	T
Guanine	G
Cytosine	C
Any nucleobase	N

Table 2.1: nucleotides with the corresponding abbreviations

In addition to the sequences themselves, FASTA also contains headers for each of the sequences, which describe the sequence on the next line. Headers start with the symbol ">".

There is a total of one mandatory and one optional parameter in the header. These parameters are separated by the first whitespace in the header sequence. The first parameter represents the name of the sequence and must be present for each FASTA sequence. The other parameter is the description of the sequence.

Since we are concerned with DNA sequences, we can focus the specification for the nucleotide sequences. By using the abbreviations for the nucleotides[REF], we can create a table of possible values in the content string.
There is a letter encoding for every possible combination of the four nucleobases, but they have been omitted due to being fairly uncommon compared to the five presented.
It is also recommended to keep the number of characters per line less than 80, but consistent throughout the file.

An example of the file format would be the following:

```
>HSBGPG Human gene for bone gla protein (BGP)
GGCAGATTCCCCCTAGACCCGCCCGCACCATGGTCAGGCATGCCCCTCCTCATCGCTGGGCACAGCCCAGAGGGT
ATAAACAGTGCTGGAGGCTGGCGGGGCAGGCCAGCTGAGTCCTGAGCAGCAGCCCAGCGCAGCCACCGAGACACC
ATGAGAGCCCTCACACTCCTCGCCCTATTGGCCCTGGCCGCACTTTGCATCGCTGGCCAGGCAGGTGAGTGCCCC
```

2.3.2 FASTQ

The FASTQ [5] format follows FASTA, but contains additional Phred[6] quality scores. The quality score is computed by the formula $Q = -10\log_{10} P$, where Q is the corresponding quality score and P is the base error probability(probability that the base is wrong). For example, a quality score of

10 would mean that the prediction is correct 90% of the times, while a score of 60 would imply that it is correct 99.9999% of the time, or one error in a million.

The outline of the file format consists of four lines. The first line starts with an "@" symbol, followed by the name of the sequence. There is no whitespace between the "@" and the sequence name. There is an option for an additional parameter, in the form of length=X where X is the length of the read, separated from the name by a whitespace.

The second line consists of the sequence itself, similar to FASTA and based on the same alphabet.

The third line contains a "+" symbol, with an optional sequence name parameter following it, without a whitespace.

Finally, the last line contains the quality scores. The scores themselves are encoded in single letter values. The quality scores in ascending order are the following. "!" represents the smallest value while "~" represents the highest.

```
!"#$%&'()*+,-./0123456789:;<=>
?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
```

This gives a total of 96 possible scores for quality. However, the quality score "#" is used for filling the quality string when the overall quality of the read is so bad it is not recommended to use it. This usually occurs when most of the bases in the sequence have a quality score of 15 or lower.

While according to the original specification, the sequence and quality strings may be split over multiple lines, it is strongly discouraged, because the splitting markers ("@" and "+") are also contained in the quality string. Such an overlap could cause errors in parsing.

The format may be similar to FASTA but is generally used for storing reads instead. This means that each sequence block in the FASTQ format will be smaller and contained on one line.

An example of this file format would be the following:

Filed name	Data Type	Description
QNAME	String	Name of the read
FLAG	Int	bitwise flag that tries to note down strand
RNAME	String	Reference sequence name
POS	Int	Leftmost position of the read
MAPQ	Int	Mapping quality value
CIGAR	String	modifications done to match reference
RNEXT	String	Reference sequence name of the next read
PNEXT	Int	Position of the next read
TLEN	Int	length of the region mapped to
SEQ	String	The sequence of the read, similar to FASTA
QUAL	String	Quality of the read, similar to FASTQ

Table 2.2: Mandatory fields of the SAM format

```
@SRR566546.970 HWUSI-EAS1673_11067_FC7070M:4:1:2299:1109 length=50
TTGCCTGCCTATCATTTTAGTGCCTGTGAGGTGGAGATGTGAGGATCAGT
+SRR566546.970 HWUSI-EAS1673_11067_FC7070M:4:1:2299:1109 length=50
hhhhhhhhhhghhghhhhhfhhhhhfffffe`ee[`X]b[d[ed`[Y[^Y
```

2.3.3 SAM/BAM

SAM[7] stands for Sequence Alignment/Map format. The format itself is TAB-delimited text, consisting of a header and an alignment section.

There are a total of 11 mandatory fileds in the file specification for a single alignment entry. These are the following:

In addition to these fields, the SAM/BAM specification also holds a number of optional fields. Such fields are commonly delimited by a two letter combination followed by a colon, the data type of the entry, followed by another colon, and the entry of the field. Due to the multitude of such fields, the ones presented below are produced by the DNA methylation mapped Bismark[8] used in this thesis as well: The SAM format is a file format readable by a human. However, recently, software has been moving on to the use of BAM, which is essentially just a binary file format of SAM. It more compact, but is not readable by a human due to its compressed nature. Luckily, the conversion between the two formats is fairly simple, since the fields contained in the files map one-to-one with each other.

An example of a SAM file field would be the following:

Filed name	Data Type	Description
NM-tag	Int	Edit distance to the reference
MD-tag	String	base-by-base mismatches to the reference
XM-tag	String	String of methylation per nucleobase in the read
XR-tag	String	Read conversion state for the alignment
XG-tag	String	Genome conversion state for the alignment

Table 2.3: Bismark fields for SAM

```
HWI-D00154:61:C2BJLACXX:4:1101:2448:2659_1:N:0: 0       chr1    2584389 255     101M
        *       0       0
TGGAGTAGTGTTTATATTTTTAGGTGAGTATGTGATAGTGTGGAGTAGTATTTATAGTTTAAGGTGAGTATCTGATAATTTGGAGTAGTAATTATATTTTT
@@@DDADDHFHHHEIBHHHIGBFGCFFBFFEHCFFGAA:CFFB@BDDAF?FHIGHDGHGHHGIIHGAHFBHGHIEGHEHGICEEG??EEAEEEEEEEEDEEC
NM:i:36 XX:Z:5C4CCC1C1CCCCC7C6C2C6C2C1CCC1C2CCC8C6C2CC5C2C2CC1C1CCCCC
XM:Z:.....x....hhh.h.hhhhx.......h......x..z......x..h.hhh.x..hhh........h..X...h..hx.....x..h..hh.h.hhhhx
XR:Z:CT XG:Z:CT
```

The only difference between the specification and the example is that the MD tag was introduced in a later version of Bismark. In this example, the MD tag has been replaced by an XX tag. This was done because the XX tag did not reflect insertions and deletions, whereas the MD tag does. The file is also split up in the example, because it is too long to contain here in any sensible way.

3 Pairwise methylation and the method

3.1 Pairwise methylation

What we define as pairwise methylation is in essence a task of binary classification[9]. Binary classification is distributing a set of elements into two groups based on a classification rule. The classification rule can be a simple mathematical function or even something the computer will have to work out on its own, given an objective function to either maximize or minimize.

A simple objective function for such a task would be to maximize the number of correct predictions. This is a value called Accuracy of the prediction. After a closer inspection, we can see that this is not a good idea, since it is dependent on the equal cardinality of classes, which does not make it a good metric. A good illustration would be one where we have less elements in one class than we classified incorrectly in the other. Such a situation would mean that predicting a constant would improve accuracy. It is also known that the amount of potential methylation sites methylated may reach up to 90%, which would create such a situation. Thus, we need better measurements for our experiments, but we are going to discuss them later.

We know that in case of a binary classification, we have a total of two possible predictions and two possible outcomes for each prediction. If we split the outcomes up with respect to our predictions, we get something what is called a confusion matrix.

A confusion matrix is a specific table used to measure the performance of supervised learning algorithms. This table contains all four possible comparisons of binary prediction and actual outcome pairs. The confusion matrix itself looks like this:

In the case we are examining, our objects to compare will be adjacent c

		Actual value		
		p	**n**	**total**
Prediction outcome	p′	True Positive	False Positive	P′
	n′	False Negative	True Negative	N′
	total	P	N	

Table 3.1: Confusion Matrix

methylation sites. The observations contained in the confusion matrix come from reads which contain both of these methylation sites. The only thing left to decide on is the prediction rule for the binary classification.

Fortunately, the rule we are interested in this context is fairly trivial. The identity function $f(x) = x$. What this means is that after observing a potentially methylated site in the genome, we predict that the next site will have the same methylation value. The purpose of the analysis is to find out the overall strength of this hypothesis. Thus, the question we are examining is: ¨do adjacent methylation sites directly affect one another?Änother goal would be to detect and pinpoint the deviations from the average behavior in methylation sites, especially in different organisms.

3.2 Algorithm

3.2.1 General Description

The algorithm is designed to find methylation in a pariwise context. What this means is that we are looking at all the potential methylation sites in a genome. However, instead of looking at them individually, we look at all the adjacent pairs of these sites. Our goal is to map reads to the genome in such a manner that two adjacent reads are contained within one read. This means that when we have two reads of which one contains a methylation status for one of the reads in the pair and the other read for the other site, we discard such data since it is not informative

Parameter name	Data type	description
context	String	methylation context
chromosomeName	String	name of the chromosome
leftmostLoctaion	Integer	location of the leftmost pair
rightmostLocation	Integer	location of the rightmost pair
TruePositive count	Integer	Count of true positive predictions
FalsePositive count	Integer	Count of false positive predictions
FalseNegative count	Integer	Count of false negative predictions
TrueNegative count	Integer	Count of true negative predictions

Table 3.2: Output of the algorithm

enough. This restricts us to use only reads which contain two adjacent methylation sites.

3.2.2 Input

As the first input parameter to the algorithm, we take a sorted SAM file with a mandatory XM field as this field contains data about methylation.
The second input parameter is the methylation context analyzed. The default value for this parameter is CG.

The third input parameter is the destination file, which is in coma separated file(csv) format. The default value for this is myfile.csv in the directory the script was run in.

3.2.3 Output

The output of the program is a coma separated file(csv) containing the fields in the following order:
An example of the output is the following:

```
cg,chr1,10589,10609,42,18,12,11
cg,chr1,10609,10617,155,0,0,10
cg,chr1,10617,10620,100,0,0,0
cg,chr1,10620,10631,102,2,56,3
```

3.2.4 Description of the algorithm

Firstly, the input file existence is checked and the contexts under examination are defined.

Secondly, the algorithm will initialize a Red-Black tree[10] which is dereferenced and re-initialized after a change in chromosome. This tree will contain each of the methylation sites for the given chromosome. The keys after which the tree is sorted are the methylation site locations and the values for these keys are two lists. The first list will contain unique identification numbers for the reads whereas the second list contains methylation values as booleans for those locations. In this context, true represents the existence of methylation and false represents a non-methylated site.

Thirdly, the algorithm will read in each of the mapped reads, line by line. The reads are then split up to parameters.Then the algorithm checks whether the current chromosome has changed. If this has occured, the algorithm writes out the information for the current chromosome and releases memory in order to manage with a minimal memory requirement. However, if the chromosome has not changed, the algorithm will store the current position within the chromosome and traverse the XM parameter of the mapped read to find methylation site locations along with their values. Then the algorithm checks whether the expected site has already been found. If so, the algorithm adds an element to both of the defined lists to express the found methylation site in the tree.

Finally, if the tree is populated and the algorithm would print out the values to a designated file, it will traverse the tree in an In-order fashion. This means that the adjacent methylation sites will be adjacent in the traversal as well. So, the algorithm will process each of the adjacent methylation pairs in the given chromosome. The processing consists of finding equal read identificators in one of the value lists. Based on these identificators, the algorithm creates a confusion matrix with the hypothesis that the two sites will have the same methylation value. These results are then appended to the output of the algorithm ,which is written to the designated file.

3.2.5 Analysis of Complexity

The analysis of complexity is carried through to convince the reader in the positive properties of the algorithm and let the reader assert, whether they are interested in using such an algorithm.
Firstly, let us denote the number of reads corresponding to a given chromosome

as N_c, where $c \in chromosomes$. This means that $\sum\limits_{c \in chrs} N_c = N$, which is the total number of reads. Let's also denote that the number of chromosomes is equal to C Such a distinction enables us to split up the problem in order to give adequate bounds to the complexity of the algorithm.

Now, we look at each chromosome separately.

For each chromosome, we initialize a Red-Black tree, which is a balanced tree structure, granting the user a guaranteed logarithmic insertion and find. The structure itself can also be traversed in an in-order fashion to receive a sorted list of elements.

Using this knowledge, we can now define an upper bound for the complexity of our task.

Let us assume that the maximal size of a read across all reads is M.

We also note that the maximal number of insertions is $N_c \cdot M$, since there is a total on N_c reads, whose maximal length is M.

We note that the worst case for inserting $N_c \cdot M$ elements occurs when all the elements are distinct. Then we would have to search for each element before adding it. Thus, instead of one logarithmic operation of finding the element and modifying its value, we have to try find the element and then add it. Thus, we can assume that the complexity of adding one element is no larger than $O(2 \log N_c \cdot M) = O(\log{(N_c \cdot M)})$. We also note that the maximal number of elements in this tree can be at most $N_c \cdot M$. So, the complexity of filling one map is $O(\sum\limits_{i=1}^{N_c \cdot M} \log_i) \leq O(\sum\limits_{i=1}^{N_c \cdot M} \log{(N_c \cdot M)}) \leq O(N_c \cdot M \cdot \log{(N_c \cdot M)})$. We have yet to include the post processing of the structure.
As for the post processing step, we traverse the methylation sites as pairs and compute the corresponding confusion matrices. The sorted traversal of all the methylation sites is with a complexity $O(N_c \cdot M)$ since our data structure already sorts the data on insertion. However, computing the confusion matrices is trickier.

We store the reads mapped to each methylation site in a list variable inside the methylation site object. What we have to find are the reads that match on both of the methylation sites. In order to achieve this, we can just sort the list containing the reads. Such a sorting can be done in $O(N_c \cdot \log N_c)$ since every read can contain the methylation site only once. After sorting, the complexity of finding equal read identifiers is $O(N_c + N_c) = O(N_c)$ because traversing both of the lists once is sufficient to find all equal elements. If done in a fashion that in case of inequality, the list pointer is moved to the next element in the list whose current element is smaller. In case of equality, both of the list pointers are moved. This gives us a total complexity of $O(N_c + N_c \cdot \log N_c) = O(N_c \cdot \log N_c)$

We can now find the complexity of processing one chromosome. That is $O(N_c \cdot \log N_c + N_c \cdot M \cdot \log(N_c \cdot M))$ since the steps are exclusive, we can add instead of multiply). For the complexity of the algorithm in general, let us assume that $\max_{cinchrom} N_c = N_x$. Now, we can say that $O(\sum\limits_{c \in chrs} (N_c \cdot \log N_c + N_c \cdot M \cdot \log(N_c \cdot M))) \leq O(\sum\limits_{c \in chrs} (N_x \cdot \log N_x + N_x \cdot M \cdot \log(N_x \cdot M))) \leq O(C \cdot N_x \cdot \log N_x + C \cdot M \cdot N_x \cdot \log(N_x \cdot M))$

Let us also note that we cannot reduce this formula any further, since it would not make sense to add bounds to the M variable. Even it it is zero, some processing is done.

3.2.6 Implementation

The result is an R package, which due to the goal to maximize performance, also requires a Java Virtual Machine(JVM) to be present in the machine it is run on. This is due to the usage of the RJava library, which enables the usage of Java inside R. Thus, all of the computations done by the package are actually written in Java and wrapped inside an R package.

The choice of Java was made based on the availability of the JVM. R was chosen as the language to host the package mainly because most of the tools dealing with methylation are presented as R packages. Thus, the motivation for using R is consistency across already existing tools.[16,17,18,19]

The package is located at[20].

4 Overview of the analysis pipeline

4.1 Pipeline

The purpose of this section is to describe the overall process of conducting an experiment with DNA methylation. It is necessary due to the complex nature of such experiments and the number of steps required to get a result, yet alone, an adequate one.

4.1.1 Introductory steps

This section will concentrate on the steps done by biologists to produce the data we are going to experiment on.

The pipeline starts with the hypothesis for an experiment. This is a clearly, unambiguously defined goal for the experiment, which defines the organism studied and the state of the organism. A good example would be changes in methylation in human breast cancer patients. The motivation for such an experiment is derived from the frequent association of cancer with the change in methylation patterns. Having defined a goal, we can proceed to gathering data. This involves taking DNA samples from the patients, and is carried out by the biologists.

Before sequencing the DNA, it must be treated with bisulfite. This causes a chemical reaction in the nucleobases, which transforms unmethylated cytosines to uracil, but leaves the methylated ones as they were. After such a conversion, the DNA is ready to be sequenced. This can be done with any sequencing technique. More modern machines are recommended due to the advantage in accuracy that they provide. Such a method is prone to error, as it relies on the quality of conversion. This means that unmehtylated cytosines that have not been treated, still read as cytosines, which means that they are counted as methylated, while in reality, they are not. The machine produces reads, which are fragments of the DNA provided to the machine, usually in FASTQ format.

4.1.2 Acquiring reads and quality control

Such sequencing runs are then stored in databases, usually in either FASTA, FASTQ, or the compressed SRA[11] file formats. Thus, the data is made accessible for bioinformaticians or enthusiasts all over the world. However, while being accessible, the reads are still not ready to be mapped to a reference genome of the corresponding organism.

DNA sequencing also produces reads that are not of high quality. These reads should not be used in an analysis. An easy tool to use for trimming the reads of the parts with a low quality score is trim-galore![12], what is an automated quality control script. Quality control and the assertion of the quality of raw/trimmed reads can be performed by FastQC[13]. This program gives the user a simple visual understanding of the current state of the reads. It also displays maximal recommended variations in the reads to assist the user even more.

4.1.3 Mapping the reads

Mapping is done with specific software. For an experiment with methylation data, it requires a few extra steps.

Firstly, since the DNA was treated with bisulfite, the reference genome needs to be transformed to a shape to cope with the difference. This means that the reference genome has to store data about methylation. Thus, the non-methylated cytosines that are converted to uracil are read as thymine by the sequencer. This means that the preprocessing of the reference genome will create another genome with all the cytosines converted to thymine. If such a conversion is done on each read as well, it will enable the program to identify methylation. The reads are then mapped to both genomes and the results merged together. For example, if a read has two possible methylation sites, of which only one is methylated, then most of the mapping overlaps, but the differences on the cytosine site mappings defines the methylation values.

This pipeline is executed by Bismark, a software for mapping bisulfite treated reads. The mapping itself is done by Bowtie, which is a general purpose aligner.

The result is a list of mapped reads. This means that the reads have a region in the reference genome where they fitted with a given number of mismatches. Since it is unreasonable to assume that an organism matches a reference in each point, we allow mismatches. They are nessecary for finding systematic annomalies as well, because else we would ignore the discrepency.

4.2 An example of an experiment

This section will describe an example workflow of for obtaining pairwise methylation data from a sequencing experiment. We will go through each of the steps, with command and code samples to compliment the process.

4.2.1 Gathering data

In order to conduct an experiment, we need a dataset. Since we are interested in methylation, the dataset has to contain bisulfite treated reads. There database of SRA reads is here[14].

As an example, let us take the dataset here[15].
We can download this with wget using the following command:

```
wget ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByStudy/sra/SRP%2FSRP028
```

Having downloaded the dataset, we have to unpack it to FASTQ format for further usage. The SRA Toolkit is used for such purposes. This can be done like this:

```
fastq-dump -I --split-3 SRR949194.sra
```

The number of files in the split parameter specifically set to the maximum value of three. This is done due to the nature of the program. If it is supposed to produce one file, it will do so, but if we specify that we only want one file and there are supposed to be more, the program will just append the other files to one file.

4.2.2 Trimming the reads

The trimming of reads can be done manually with FastQC[12], or automatically with trim galore[13].

It is clearly shown on the image how one should approach trimming. Removing everything that touches the red areas and consider whether they a lower mapping percentage but potentially more reads for the yellow region.

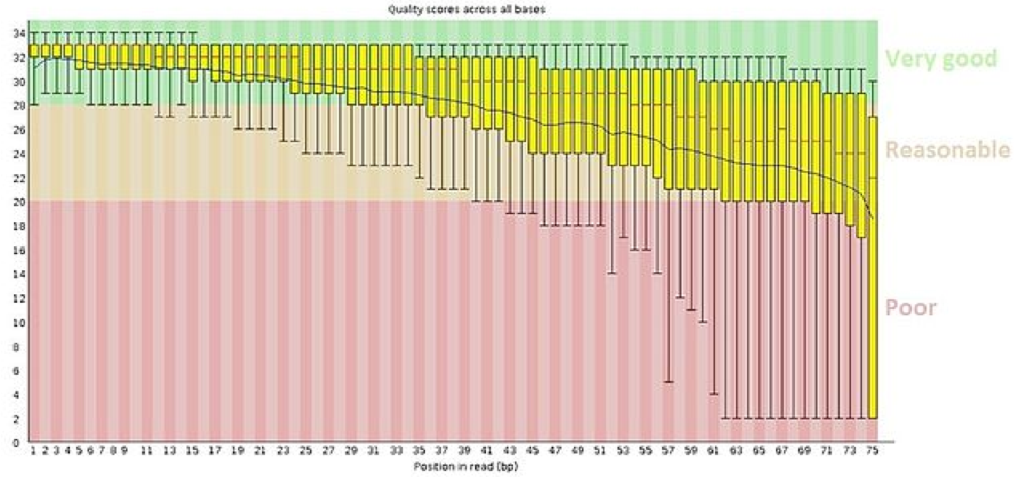

Figure 4.1: Screenshot of FastQC

4.2.3 Mapping reads to the genome

After trimming the reads, we should prepare our genome for the mapping process. With Bismark, this can be done with:

```
bismark_genome_preparation --path_to_bowtie /shared_group/software/aligners/bowtie
--verbose  /data/hg19
```

After the genome preparation has been executed, we can map the reads to the genome. For single stranded reads, we specify one input file, but for the double stranded ones, we speficy two files, one for each strand:

```
.bismark -n 3 -l 50 --path_to_bowtie /shared_group/software/aligners/bowtie
~/data/hg19 -1 ~/SRR949194_1.fastq -2 ~/SRR949194_2.fastq
```

Here the n parameter corresponds to the number of mismatched allowed(legal values are from 0 to 3). and the -l parameter corresponds to the number of bases in the high quality end of the read.

4.2.4 Extracting pairwise methylation

Finally, after preparing mapping bisulfite treated reads, we can get the pairwise methylation values. For this, we import the described R package and call producePairwiseMethylation, with a possible of three parameters. The first parameter is mandatory and it is the SAM file under analysis. The other two parameters are context names and the output file name. These parameters can be omitted as default values are provided.

This step yields the confusion matrices that can be analyzed by the various ways discussed below.

5 Analysis and Metrics

5.1 Metrics

5.1.1 Matthew's Correlation Coefficient

Matthew's Correlation Coefficient(MCC)[21] is a statistical measure used for the assertion of binary classifications. Our initial hypothesis also considered a task of binary classification to either methylated or unmethylated, based on a previous observation. This implies that the usage of such a metric is adequate for assessing the confusion matrices produced.

Another important property of MCC is that it does not suffer from the accuracy paradox. This is important, since varying from organism to organism, we can expect an average methylation up 90% across the whole organism. This implies that the dataset might be inherently biased.

Matthew's Correlation Coefficient itself is represented by the following formula,

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}$$

The range of the values is $[-1, 1]$ where the extremal values represent a strong correlation and 0 represents no correlation.

5.1.2 Linkage Disequilibrium

The other metric we will explore in greater detail is Linkage Disequilibrium.[22]

Linkage Disequilibrium is a measure of non-random association between loci. In our case, the loci represent the positions of pairs of adjacent methylation sites. It makes sense to use this measure in our case because the measure is developed for the analysis of similarity between loci.

Linkage Disequilibrium is calculated on frequencies, so the matrix of occurrences needs to be converted to a matrix of frequencies. This can be done by dividing each element in the matrix with the number of total occurrences. Since we need allele frequencies instead of the confusion matrix frequency values, we convert them using the following formulas. These values are calculated as follows: Let $X = TP + FP + FN + TN$ or in other words, X is the count of observations. The frequencies are computed by dividing the field with the number of total observations. $TP_f = \frac{TP}{X}$. The same is done for all the other elements as well.
Now we can define individual event frequencies like follows:

$$\begin{aligned} p_1 &= TP_f + FP_f \\ q_1 &= TP_f + FN_f \\ p_2 &= FN_f + TN_f \\ q_2 &= FP_f + TN_f \end{aligned}$$

Here, the p-s represent the first observation while the q-s represent the second one. Also, a subscript number one represents methylation while a subscript number 2 represents the absence of methylation. Follwing this, we can define Linkage Disequilibrium as follows:

$$D = TP_f - p_1 \cdot q_1$$

However, this measure is not very informative since it is dependent on the number of alleles. A common method to get rid of this problem is to normalize D by dividing it with the theoretical maximum.

$$D' = \frac{D}{D_{max}}$$

Where $D_max = \min(p_1 \cdot p_2, q_1 \cdot q_2)$ when $D < 0$ and
$D_max = \min(p_1 \cdot q_2, q_1 \cdot p_2)$ when $D > 0$
When $D = 0$, the coefficient is also 0. This gives us a measure in the range $[0, 1]$ where 0 corresponds to total Linkage Equilibrium or in other words, represents that the methylation of the sites is completely independent of one another, while a value of 1 represents complete dependence.

5.2 Aggregation of data

Another way to make sense of the data would be to use these metrics but aggregate the data to one big confusion matrix. Instead of giving specific information about a certain pair of methylation sites, such aggregations can give insight on the behaviour of methylation on a wider scale. For example, if we would aggregate the values by just summing the matrices, we would get a matrix that corresponds to all the methylation pairs within that region. Running a correlation test on such a matrix would yield us a value that corresponds to the average pairwise methylation in that region. This is possible, since the correlation tests discussed have a normalized range of values and don't exhibit biases caused by the number of elements nor the proportions of positive and negative classifications.

6 Further Work

This chapter discusses future approaches to dealing with pairwise methylation and analysis. The goal is to give insight to what this thesis contributed to and what possibilities this opens up. Currently, the topic of pairwise methylation in orgainsms is still relatively unknown. This article describes the now known notion of methylation islands inside organisms, which was the initial motivator for exploring this topic. Methylation islands are essentially regions within the genome with a high percentage of possible methylation sites being methylated. This poses a hypothesis that maybe methylated sites tend to be near each other. Namely, next to each other, in such a manner that transitions between methylation values are uncommon. Further work would also include comparing different pairwise methylation profiles. This could help identify diseases or mutations based on methylation profiles and lead to the association of methylation profiles with these mutations. Much like what is currently studied in genetics.

Another possibility would be to associate genetic mutations with changes in the methylation profiles.

Amongst other thing, it would be interesting to try out different classification laws or even classifiers. Such experiments would require some prior knowledge about pairwise DNA methylation, which we currently lack. Thus, this would be a good thing to test when we know more about methylation in a pairwise context. Currently, we even lack a base measurement to compare these methods with.

Unfortunately, these ideas by the author are purely hypothetical, because the topic of pairwise methylation is still very much unexplored.

7 Conclusion

The purpose of this thesis was to provide adequate insight on methylation data analysis with a practical application of producing an R package to extract methylation scores of methylation site pairs. The thesis also meant to give an introduction to methylation analysis with and give insight about analyzing pairwise methylation with various correlation measures. The result of this thesis is an R package designed to carry out a step in the analysis pipeline.

The thesis firstly introcudes concepts of DNA and methylation. Then, the notion of pairwise methylation is introduced with motivation why such an analysis would be beneficial. The chapter ends with an algorithm to obtain pairwise methylation results. The third chapter provides a general outline of a methylation analysis pipeline along with an example run of this pipeline.
The fourth and the final chapter gives insight on how to interpret the pairwise methylation data, introducing Matthew's Correlation Coefficient and Linkage Disequilibrium along with an alternative take on the data as a whole. Preliminaries contain example codes that can be used to interpret the results.

8 References

[1] DNA http://en.wikipedia.org/wiki/DNA [Online; accessed 14-May-2015]

[2] DNA Methylation http://en.wikipedia.org/wiki/DNA_methylation [Online; accessed 14-May-2015]

[3] DNA methylation detection: Bisulfite genomic sequencing analysis http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3233226/[Online; accessed 14-May-2015]

[4] Nomenclature for Incompletely Specified Bases in Nucleic Acid Sequences http://www.chem.qmul.ac.uk/iubmb/misc/naseq.html[Online; accessed 14-May-2015]

[5] FASTQ Format Specification http://maq.sourceforge.net/fastq.shtml [Online; accessed 14-May-2015]

[6] Phred - Quality Base Calling http://www.phrap.com/phred/ [Online; accessed 14-May-2015]

[7] Sequence Alignment/Map Format Specification https://samtools.github.io/hts-specs/SAMv1.pdf [Online; accessed 14-May-2015]

[8] Bismark http://www.bioinformatics.babraham.ac.uk/projects/bismark/[Online; accessed 14-May-2015]

[9] Evaluation of binary classifiers http://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers[Online; accessed 14-May-2015]

[10] Red-black tree http://en.wikipedia.org/wiki/Red-black_tree [Online; accessed 14-May-2015]

[11] SRA Knowledge Base http://www.ncbi.nlm.nih.gov/books/NBK158900/ [Online; accessed 14-May-2015]

[12] Wrapper tool for Cutadapt and FastQC, Trim Galore! http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/[Online; accessed 14-May-2015]

[13] A quality control tool for high throughput sequence data http://www.bioinformatics.babraham.ac.uk/projects/fastqc/[Online; accessed 14-May-2015]

[14] Database of SRA reads ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByRun/sra/SRR/SRR949/ [Online; accessed 14-May-2015]

[15] Read file used for the example ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByRun/sra/SRR/SRR949/SRR949194/ [Online; accessed 14-May-2015]

[16] Creating R Packages: A Tutorial http://cran.r-project.org/doc/contrib/Leisch-CreatingPackages.pdf [Online; accessed 14-May-2015]

[17] Writing R Extensions http://cran.r-project.org/doc/manuals/r-release/R-exts.html [Online; accessed 14-May-2015]

[18] rJava https://www.rforge.net/rJava/ [Online; accessed 14-May-2015]

[19] The R Project for Statistical Computing http://www.r-project.org/ [Online; accessed 14-May-2015]

[20] PairwiseMethylation, Github https://github.com/Soulyyy/PairwiseMethylation/[Online; accessed 14-May-2015]

[21] Matthews correlation coefficient http://en.wikipedia.org/wiki/Matthews_correlation_coefficient [Online; accessed 14-May-2015]

[22] Linkage disequilibrium http://en.wikipedia.org/wiki/Linkage_disequilibrium [Online; accessed 14-May-2015]

[23] Create colonCancerWGBS https://github.com/genomicsclass/colonCancerWGBS/blob/master/scripts/createObject.Rmd[Online; accessed 14-May-2015]

[24] SAMtools http://samtools.sourceforge.net/ [Online; accessed 14-May-2015]

Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, Hans Peeter Tulmin(08.05.1993),

1. herewith grant the University of Tartu a free perimit(non-exclusive licence) to:

   (a) reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   (b) make available to the public via the web environment of the University of Tartu, including via DSpace digital archives, as of 26.06.2016 until expiry of the term of validity of the copyright,

   "Pairwise DNA Methylation Analysis", supervised by Balaji Rajashekar,

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property right or rights arising from the Personal Data Protection Act.

Tartu, 14.05.2015