Comenius University Bratislava

Faculty of Mathematics, Physics and Informatics

Correcting read count bias in aneuploidy prediction

Bachelor's Thesis

2015

Alena Damková

Comenius University Bratislava

Faculty of Mathematics, Physics and Informatics

Correcting read count bias in aneuploidy prediction

Bachelor's Thesis

Study programme:	Computer Science
Study field:	2508 Computer Science
Study department:	Department of Computer Science
Supervisor:	Mgr. Jaroslav Budiš

Bratislava, 2015

Alena Damková

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Alena Damková
Study programme:	Computer Science (Single degree study, bachelor I. deg., full time form)
Field of Study:	9.2.1. Computer Science, Informatics
Type of Thesis:	Bachelor´s thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Correcting read count bias in aneuploidy prediction

Aim: Prediction of specific genetic disorder, aneuploidy, is complicated by systematic bias that arise in obtaining input data. The aim of this work is to implement various methods for its correction and test their impact on the quality of diagnosis.

Supervisor:	Mgr. Jaroslav Budiš
Department:	FMFI.KI - Department of Computer Science
Head of department:	doc. RNDr. Daniel Olejár, PhD.
Assigned:	13.10.2014

Approved: 28.10.2014 doc. RNDr. Daniel Olejár, PhD.
Guarantor of Study Programme

.....
Student

.....
Supervisor

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Alena Damková
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.2.1. informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Correcting read count bias in aneuploidy prediction
Korekcia skreslenia počtov čítaní pri predikcii aneuploidií

Cieľ: Diagnostiku špecifických genetických porúch, aneuploidií, komplikujú systematické skreslenia, ktoré vznikajú pri získavaní vstupných dát. Cieľom práce je implementovať viaceré metódy na korekciu a otestovať ich dopad na kvalitu diagnostiky.

Vedúci: Mgr. Jaroslav Budiš
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: doc. RNDr. Daniel Olejár, PhD.

Dátum zadania: 13.10.2014

Dátum schválenia: 28.10.2014

doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Acknowledgments

I would like to thank my supervisor Mgr. Jaroslav Budiš for suggesting this topic and for his valuable help and guidance.

Abstract

Aneuploidy is a genetic mutation causing severe damage to an affected individual. Non-invasive methods of aneuploidy prediction estimate the presence of aneuploidy in a DNA sample by processing outcome of DNA sequencing. An output of the analysis is however highly influenced by GC content bias, caused by uneven composition of DNA.

We implemented a method for aneuploidy prediction and introduced current trends in correction of GC bias. Various methods of GC bias correction were designed and implemented on a set of DNA samples. We compared the results of proposed approaches and evaluated the most efficient method of GC correction. By combination of different approaches we improved results of diagnosis compared to standard correction methods.

Keywords: aneuploidy prediction, trisomy 21, GC correction, Loess correction, within-sample correction.

Abstrakt

Aneuploidia je genetická mutácia spôsobujúca vážne postihnutie. Neinvazívne metódy predikcie aneuploidii stanovujú prítomnosť aneupolidie v DNA vzorke spracovaním výsledkov DNA sekvenovania. Avšak výsledok tejto analýzy je do značnej miery ovplyvnený GC skreslením spôsobeným nepravidelnou kompozíciou DNA.

Implementovali sme metódu predikcie aneuploidii a predstavili súčasné trendy korekcie GC skreslenia. Rôzne metódy korekcie GC skreslenia boli navrhnuté a implementované na súbore DNA vzoriek. Porovnali sme výsledky navrhnutých metód a vyhodnotili najefektívnejšiu z nich. Kombináciou rôznych prístupov sme zlepšili výsledky diagnózy v porovnaní so štandardnými metódami korekcie.

Kľúčové slová: predikcia aneuploidie, trizómia 21, GC korekcia, Loess korekcia, korekcia v rámci vzorky.

Contents

List of Figures

List of Tables

Introduction

Aneuploidy is genetic mutation, that causes severe damage to an individual.

Modern medicine allows us to predict the presence of aneuploidy in prenatal state before an affected child is born. There are invasive and highly reliable methods of prediction, however these methods carry a certain risk for mother as well as for the unborn child. To overcome possible complications caused by invasive testing procedures non-invasive methods are being developed. In non-invasive methods presence of aneuploidy is estimated from blood sample of a pregnant woman, so there is no risk for a mother or her child.

Non-invasive aneuploidy prediction utilises data from DNA sequencing: the sequence of maternal DNA, containing a portion of DNA of the fetus, is obtained in small pieces – sequencing reads, that are assembled back together to reveal the original DNA sequence.

Aneuploidy prediction is based on number of sequencing reads obtained from DNA sequencing. Unusually high number of sequencing reads in certain regions of DNA indicates presence of aneuploidy. However, lately it has been shown, that number of reads obtained from DNA sequencing depends to large extent on the composition of the DNA sequence itself. DNA sequence consists of four different bases and the percentage of certain bases (cytosine, guanine) in different parts of DNA sequence affects the outcome of DNA sequencing. This phenomenon is referred to as GC content bias and methods trying to decrease its influence are called GC correction.

The aim of our work is to design, implement and compare various methods

of GC correction. The methods are tested on a set of samples, in some of which an aneuploidy is present. The best method would be used in a desktop application used by doctors to diagnose aneuploidy in high-risk pregnancies.

In the first chapter we introduce biological background of this problem, describe the process of non-invasive aneuploidy prediction and state aims of our work. At the beginning of second chapter the standard method of aneuploidy prediction is clarified. Common approaches to GC correction are introduced as well as the methods, that we designed. The third chapter is dedicated to the implementation of GC correction methods. The used technologies are described at the beginning of the chapter. The rest of the chapter introduces and describes implemented classes. The last chapter summarises accuracy of implemented methods.

Chapter 1

Biologic background and problem statement

In this chapter we define some terms crucial for further understanding and introduce concepts, on which the next chapters are built. We provide basic overview of the way genetic information is stored and encoded, explain the condition of aneuploidy and provide basic information on DNA sequencing and GC content bias. At the end of the chapter we state the aims and purpose of our work.

1.1 Genetic information

Genetic information determines functioning and features of all the cells in our body. Eukaryotic organisms (including humans) have their genetic information stored in few molecules of DNA in cell nucleus.

DNA or deoxyribonucleic acid is a structure formed by two polynucleotide strands of opposing polarity forming a double helix. Polynucleotide strands have two ends with different chemical characteristics marked as 5' end for beginning and 3' end for the end of a strand. They consist of large number of nucleotides. Nucleotides of each strand are bound together by hydrogen bonds and thereby bounding the two strands. Nucleotides consist of phos-

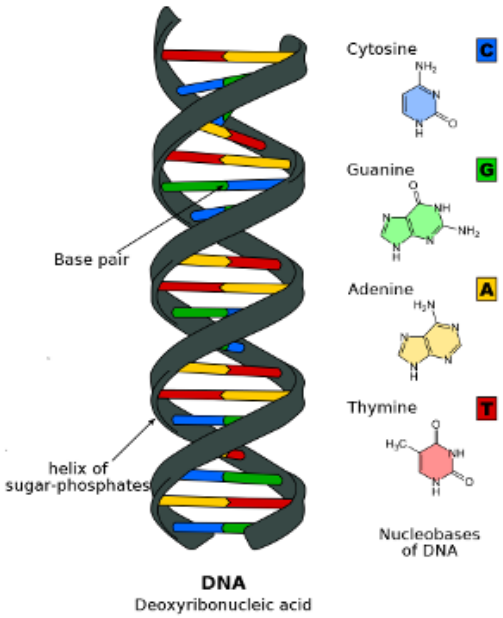

Figure 1.1: DNA structure

phate group, sugar called deoxyribose and a base containing nitrogen. There are four different bases occurring in human DNA, two of them are purine bases – adenine, guanine – and the other two pyrimidine bases – cytosine and thymine. The only difference among nucleotides are their bases, that is why we will not strictly differentiate between the terms nucleotide and base in the further text.

Binds between bases of two polynucleotide strands normally follow complementarity rule: Specific purine bases form binds only with specific pyrimidine bases (adenine with thymine, guanine with cytosine). Figure 1.1 [wikipedia.org] illustrates the structure of DNA and binds formed according to complementarity rule. This fact allows us to work with sequence of bases representing only one strand of DNA, as the other one can be easily determined using the complementarity rule. The order of nucleotides (bases) in DNA encodes the genetic information.

DNA molecules in cell nucleus form specific amount of chromosomes.

Amount, shape and size of chromosomes differs among species of organisms, but it does not differ significantly among individuals of the same species.

Most of cells forming our body, also called somatic cells, contain two sets of chromosomes and are referred to as diploid. Only gametes used during reproduction process have one set of chromosomes and are referred to as haploid. During fertilization two gametes fuse with each other, each of them passing one haploid set of chromosomes to the fetus, which combined create a diploid cell with two sets of chromosomes and thereby present the fetus with full and valid genetic information.

Somatic cells of humans contain 46 chromosomes (two sets of 23 chromosomes). First 22 pairs are called autosomes and are labeled by a number from 1 to 22. The last pair is called sex chromosomes and determines sex of the holder. First of the pair is chromosome X and the second one is chromosome Y or chromosome X depending on whether the holder is male or female, respectively. [16]

1.2 Aneuploidy

Aneuploidy is a chromosome mutation, where the number of chromosomes in a cell nucleus differs from a regular diploid or haploid number. Aneuploid cell can possess more or less chromosomes than a regular cell. In most cases aneuploids differ by only one chromosome. [5]

This kind of chromosome mutations is usually caused by errors occurring during the process of cellular division, when the chromosome pairs do not separate. [4] Aneuploidy concerns autosomes as well as sex chromosomes. The balance of genes and their functions in a regular cell is very delicate, therefore an extra copy of a chromosome, holding hundreds of genes causes severe damage, which is in most cases not compatible with life. [12]

Specific cases of aneuploidy are named according to the amount of missing or extra chromosomes: monosomy, trisomy, tetrasomy... The most common human aneuploidy is trisomy – a condition when tree copies of a specific

chromosome instead of regular two are present in a cell nucleus (the overall chromosome number is 47 instead of 46). [5]

The only human autosomal trisomies intervening in live births are trisomy 13, 18 and 21. Trisomy 13 occurs in one out of 19 000 live births and causes a disorder called Patau syndrome. Trisomy 18 occurs in one out of 8 140 live births and is a cause of Edwards syndrome. [11] Both of these conditions result in most cases in death during the first year of life. [5] The most common is trisomy 21 also known as Down syndrome, present in one out of 800 live births.[11] Figure 1.2 [12] shows the presence of three copies of chromosome 21 in chromosome set of a female patient with Down syndrome. Mean life length expectancies for patients with this condition are 17 years, but some of them can live over 40 years. [5] Chromosome 21 has the smallest number of protein-coding sequences out of all autosomes, therefore also the number of extra genes is smaller and the damage caused is less significant, which is assumed to be a reason why only trisomy 21 is compatible with a longer life span. The risk of pregnancy with trisomy 21 rises essentially with the increasing age of mother. Therefore pregnant women over 35 are offered fetal chromosome analysis (undergoing amniocentesis or chorionic villus sampling). Also a connection between higher paternal age and occurrence of trisomy 21 was shown.

Trisomies of sex chromosomes are occurring in a portion of live births, however the damage caused is significantly less severe than the damage caused by trisomy of autosomes. This can be explained by characteristics of sex chromosomes. In our work we focused on human trisomies, particularly on trisomy 21, as it appears with the highest frequency.

1.2.1 Aneuploidy prediction

In past few years a significant progress has been made in the area of aneuploidy prediction. In cases of high-risk pregnancies (women after 35) clinical prenatal tests for common aneuploidies are offered. The most reliable testing methods are invasive - amniocentesis or chorionic villus sampling.

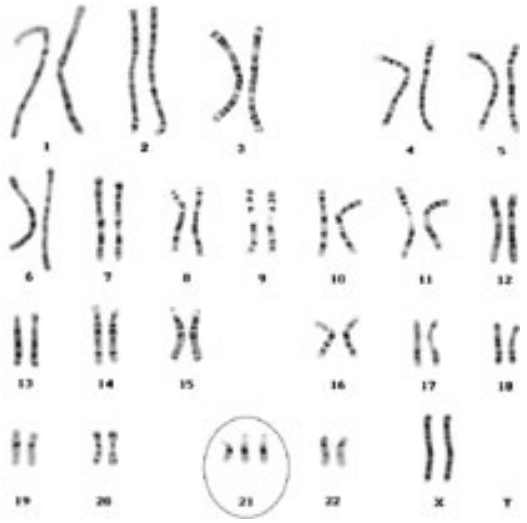

Figure 1.2: Trisomy 21

However these methods carry a risk of harming the fetus or in 1% of cases lead to miscarriage. Furthermore they cause more inconvenience and discomfort for pregnant women. Therefore research has focused on non-invasive methods of aneuploidy testing. These methods are not as reliable as invasive methods, hence in case of positive results, it is recommended to obtain a confirmation by one of the invasive methods.

In the non-invasive prenatal testing a sample of maternal blood is collected. A cell-free DNA is obtained from blood plasma. According to the research of Lo et al [10] 3-13% of maternal plasma are comprised of fetal DNA. In the further process DNA sequencing followed by computational analysis is applied in order to obtain predictions for selected types of aneuploidy.

1.3 DNA sequencing

Sequencing is a process of obtaining genetic information encoded by order of bases in DNA. DNA can be defined as set of words over the alphabet $\Sigma_D = \{\text{A,C,T,G}\}$.

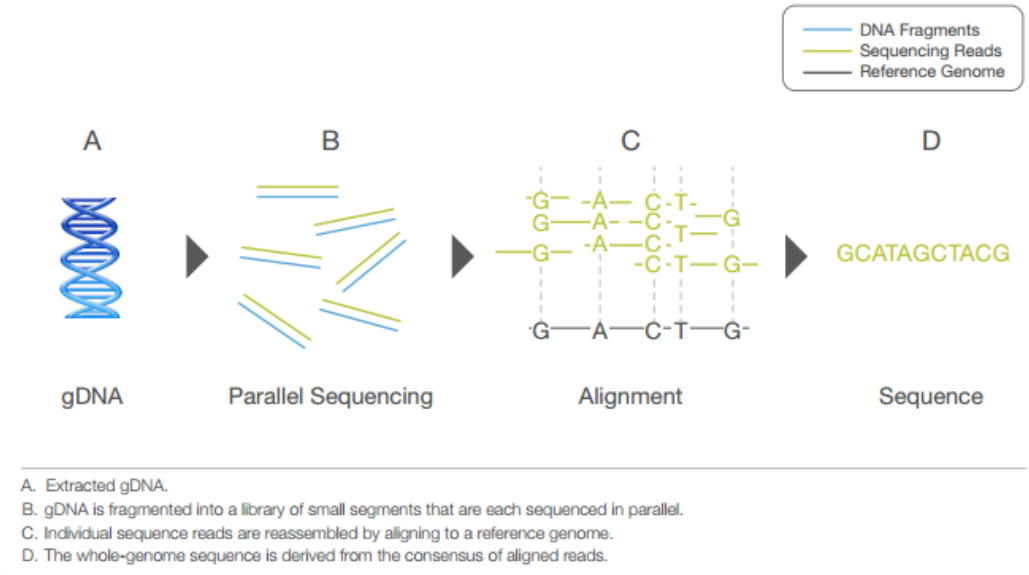

Figure 1.3: The process of DNA sequencingt

The first techniques of DNA sequencing were established in 1970s and a reference of the whole human genome was obtained in 2003. The most recent sequencing methods, also called new generation sequencing, are constantly decreasing the time necessary for determination of human genome, lowering the costs and trying to make DNA sequencing commercially available and affordable. Multiple commercial and research methods are used by different companies and research centres in order to obtain the structure of various DNA samples. Samples that we used later in our experiments were sequenced by a major sequencing company Illumina.

As human genetic information is encoded in over 3 billion bases, no available sequencing technology is able to evaluate all of them in one piece. The DNA sequence is therefore randomly cut into smaller pieces, in which the order of bases is easier to determine. These small fragments are sequenced from one side, the base-order is obtained in pieces of text information, called sequencing reads, and mapped back to the reference of human genome. Genome of each individual differs in certain bases from genome of any other

person. When the reads are mapped back to the human genome reference, slight differences between the sample ganome and human reference are allowed. Only reads that can be uniquely mapped to only one position in genome are usually considered for further analysis. When the full set of sequencing reads is mapped to the genome, the base-sequence for each chromosome is revealed. Tho process of obtaining of the order of bases in DNA is illustrated in Figure 1.3 [6]. To ensure high reliability of the outcome, more strands of DNA are used for sequencing. All of them are cut randomly into smaller pieces, which results in a massive amount of randomly overlapping reads and thereby more reads mapped to every specific position. The allignment of overlapping reads to reference sequence is illustrated in Figure 1.4 [6] This decreases the probability of erroneous base-mapping. The average number of reads mapped to each position in genome is called read coverage.

If similar repetitive subsequences are present in DNA a problem arises, as the reads can not be mapped to this position in a unique way. To overcome this complication a fragment of DNA can be sequenced from both sides. The outcome of this approach are pair-end reads (mate pairs) - a certain number of bases is determined from each side of the fragment, allowing some bases in the middle of the fragment to be unknown in this read. An example of pair-end reads is shown in Figure 1.4.

A DNA sequence can be evaluated at different level of accuracy, which is labeled as sequencing quality. Sequencing quality 10 stands for base call accuracy of 90%, expressing the probability, that 1 in 10 bases was evaluated incorrectly. The Illumina sequencers produce majority of sequencing reads with quality higher than 30, which stands for 99.9% base call acuracy, meaning that an incorrectly determined base should occur once in 1000 bases. [7]

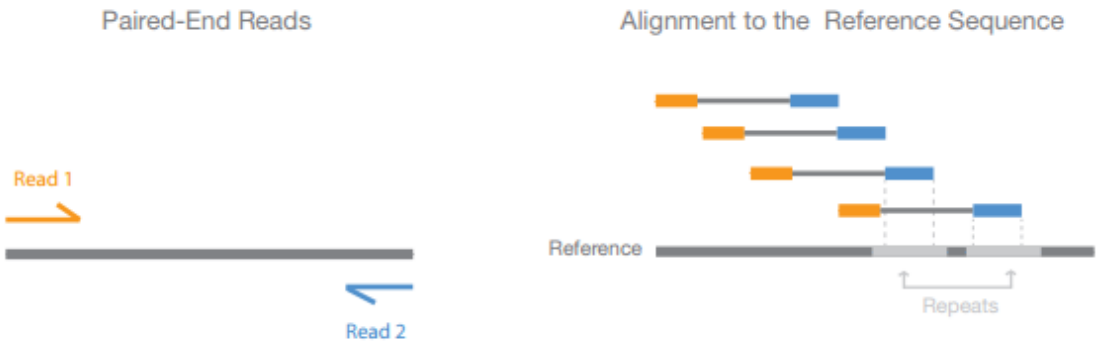

Figure 1.4: Pair-end sequencing and alignment

1.4 GC content bias

Benjamini and Speed [1] defined GC content bias as a dependence between content of guanine and cytosine bases found in Illumina sequencing data and read coverage. In other words, GC content in a sequence influences how many reads mapped to this sequence will be obtained from DNA sequencing. This fact can corrupt results of further analyses of genome, for instance in the case of methods based on observing the abundance of some bases, the only cause of difference in results among various samples can be GC content bias.

A roughly linear GC content bias was first observed by Dohm et al (Substantial biases in ultra-short read data sets from high-throughput DNA sequencing) in 2008. Later it was shown, that GC content bias is not linear. The bias is not consistent among different DNA samples or even among repeated experiments. Figure 1.5 [1] illustrates different behaviour of number of sequencing reads mapped to segments with various GC content among two sequencing libraries of the same sample (the GC content bias is different in each sequening run). It is known, that GC bias depends on the base content of a whole DNA fragment, not just the sequenced read. Both GC- and AT-rich fragments are affected by this effect and underrepresented in the results

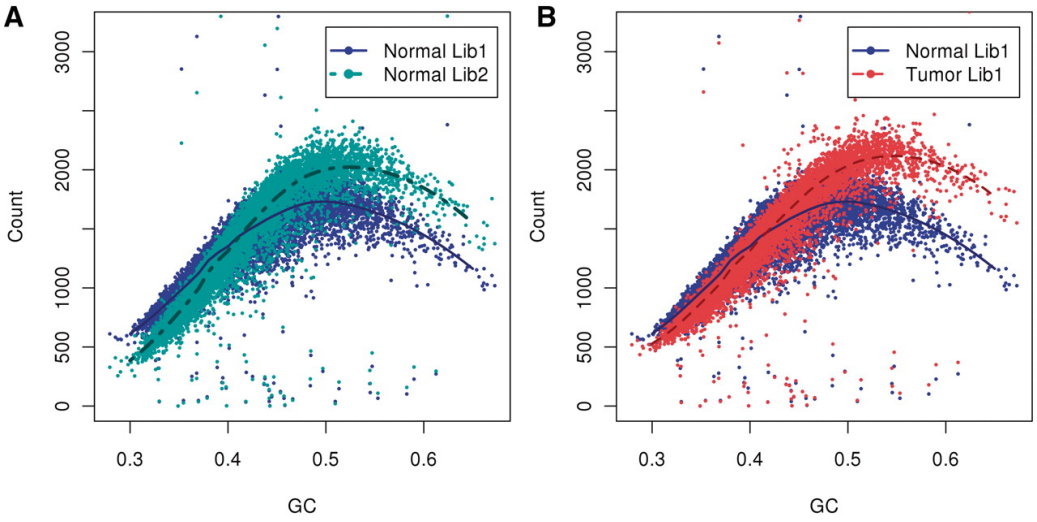

Figure 1.5: Numbers of reads mapped to fragments with different GC content

of sequencing.

Since its discovery GC bias has been subjact to many studies trying to proove its origin and suggesting methods for removing the GC bias and producing unbiased data. Untill today the most efficient and suitable method of GC bias correction was not agreed upon. In our work we focused on two particularly interesting methods: Loess correction and within-sample comparison scheme.

1.5 Problem statement

As already mentioned, in the process of aneuploidy prediction a number of reads mapped to each segment of genome is crucial. However, the number of reads is affected by CG content bias and depends on the DNA sequence itself. This parameter causes biased results. As the positive answer in aneuploidy prediction might lead to the pregnant woman undergoing an abortion, it is essential, that the results are as precise as possible. Therefore in our work we focus on suggesting different methods, inspired by recent research

and publications, reducing GC content bias. We aim to implement different methods of reducing GC content bias and examine the effectiveness of those methods alone as well as combined.

When a DNA sample is tested for presence of aneuploidy a final number, referred to as z-score, is computed for the sample. If the z-score is higher than 3, the sample is expected to be trisomic and non-trisomic in the other case. However if the result lies in a range between 2.5 and 4.0, the presence of trisomy is not clear as such results might be a consequence of some inaccuracy in sequencing (e.g. GC content bias).

By applying methods of GC correction to sequencing results, we expect z-scores of the DNA samples to converge to values, which can clearly determine presence of aneuploidy.

Chapter 2

GC correction

In this chapter we describe the method of aneuploidy prediction and introduce the methods of GC correction, that we implemented.

We adopted two common approaches to GC bias correction: Loess and within-sample correction. In first stage a Loess correction was applied. Three different methods of within-sample correction were designed and applied to Loess corrected as well as uncorrected data. The flow chart in Figure 2.1 illustrates the order of applied methods.

2.1 Sequenced samples

For implementation and testing of our methods we used a set of different samples of sequenced genomes, consisting of trisomic and non-trisomic samples. A subset of diploid samples was selected as training samples. These were used for estimation of characteristic behaviour of diploid samples. The rest of the samples were used as testing samples.

Each sample was represented by a file in SAM format, which contains information about read pairs from the sample mapped to various positions of the reference genome (we describe SAM file format in section 3.3).

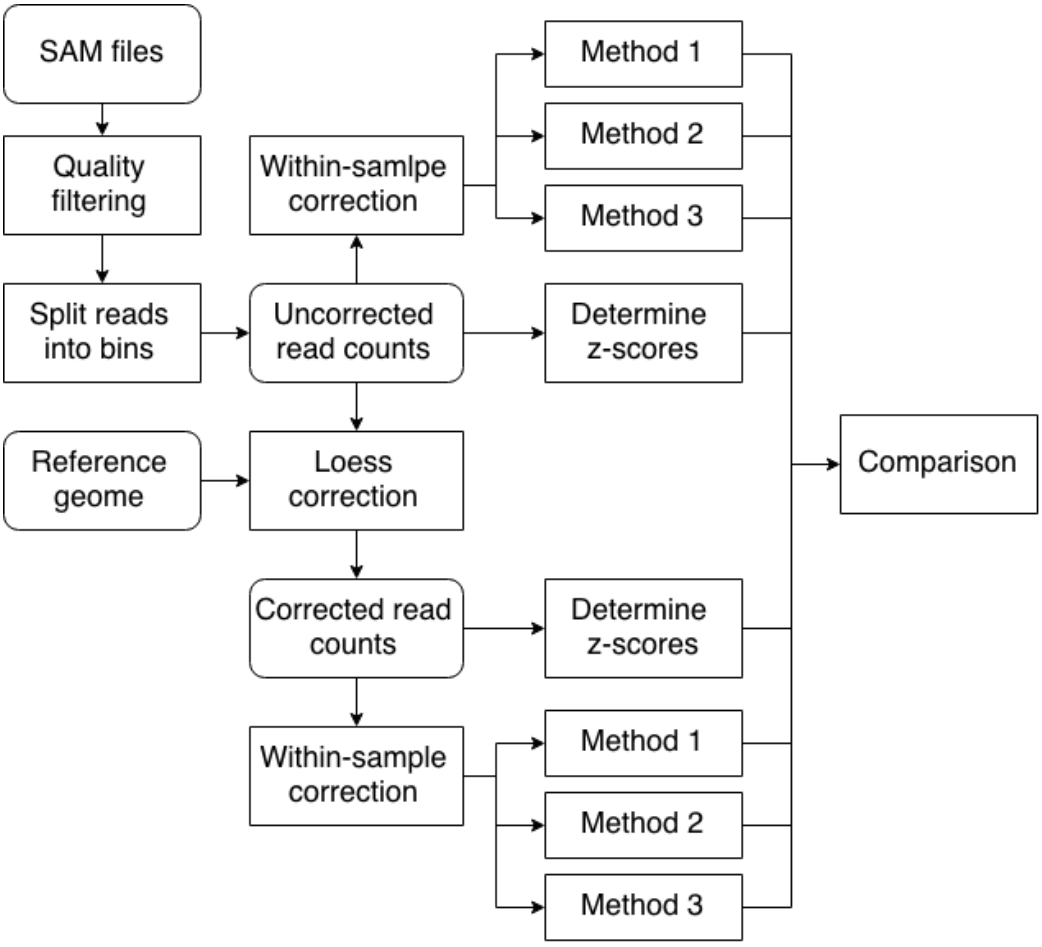

Figure 2.1: Executed methods of GC correction

2.2 Quality filtering

For a read pair to be considered for further analysis, both reads of the pair had to be mapped to the same chromosome, both with mapping quality greater or equal to 40 and the insert size (number of bases between end of the first and end of the second read) had to be lower than 10 000. Segments of DNA, in which any indeterminable bases (positions for which an exact base was not obtained in reference genome) occure in reference genome, were omitted in further analysis.

2.3 Non-invasive aneuploidy prediction

As we already mentioned in section 1.2.1, we focus on predicting aneuploidy from a blood sample of pregnant women. In the process of prediction fetal DNA is extracted from this sample, sequencing methods are applied and sequencing reads are obtained. These reads are mapped to specific positions in reference human genome. The results of this primary analysis are stored in .SAM files, each of the tested subjects being represented by one file. In order to predict trisomy, sequencing reads are sorted depending on the chromosome, to which they were mapped. The sum of reads mapped to each chromosome is computed. If the sample is trisomic the ratio of reads mapped to trisomic chromosome should be higher than in diploid sample. Chromosome ratio was defined as

$$r_i = \frac{s_i}{s_{ch}} \tag{2.1}$$

where s_i is sum of reads mapped to chromosome i and s_{ch} is sum of all the reads, that were uniquely mapped to any autosome (chr1 - chr22).

A set of diploid samples is essential in order to determine usual chromosome ratios. In our work we used a set of non-trisomic samples for this purpose. The chromosome ratio was estimated only for chromosome 21 of each sample. This was due to the fact that not many samples with other

trisomies are available and therefore it would be difficult to test our methods on trisomies other than 21. However, the same process can be followed for any other chromosome.

The set of chromosome ratios of unaffected samples follows normal distribution. Normal distribution is a statistical distribution of continuous random variables denoted $N(\mu, \sigma^2)$, where μ represents mean of the distribution and σ its standard deviation. The 3-sigma rule applies to normal distribution, stating that 99.7% of the values lie no further than 3 standard deviations from the mean. We use this theorem for assigning a final z-score to a sample.

Z-score of chromosome 21 was computed for each testing sample with following equation:

$$z = \frac{r_{21} - \mu}{\sigma} \tag{2.2}$$

where μ stands for mean of the normal distribution of reference euploid samples and σ stands for its standard deviation. If the z-score of given testing sample is greater or equal to 3, it is evaluated as trisomic. Z-score of euploid sample should be in interval of $[-3, 3]$, but since we were examining only trisomy 21, the condition of $z < 3$ is sufficient for our purposes.

2.4 Loess GC correction

Most of current GC corrections follow a common pattern based on the idea, that fragments of DNA with similar percentage of GC content should behave alike. Reference genome is divided into segments of selected size called bins and relative abundance of cytosine and guanine nucleotides (GC content), expressed in percentage, is determined for each of them. For each sequenced sample a curve describing the conditional mean read count per GC value is estimated. The resulting GC curve implies predicted read count for each bin based on GC content of the bin. [1]

Among others Liao et al. [9] used this approach to GC correction. In order to obtain GC curve they applied local weighted regression to mean read counts against GC content. We followed similar approach.

The reference genome was divided into bins of wished size and the percentage of G and C bases was estimated for each bin. We rounded percentual GC contents to one decimal place.

Quality filtering described in section 2.2 was applied to read pairs of a sample. For each bin in the testing sample a number of reads mapped to this bin was determined.

For each possible GC content a list of bins with the respective GC content was assigned. The mean number of reads mapped to respective bins was estimated for each GC content:

$$m_p = \frac{\sum_{i=1}^{n} (r_i^p)}{n_p} \tag{2.3}$$

where p is percentage of GC content, n_p is number of bins of given size with $p\%$ of GC content, r_i^p is number of reads mapped to bin i with GC content p. A weight was assigned to each GC content depending on how many bins had corresponding content:

$$w_p = n_p \tag{2.4}$$

Weighted loess regression was applied to the list of means m_p with weights w_p. Local weighted regression (loess) estimates regression surface through a multivariate procedure fitting a function of independent variables locally and in a moving fashion. Some m_p values can be significantly higher or lower than the rest if only few bins have GC content p. Loess regression smooths the values of means m_p and removes local extremes, caused by insufficient read coverage for some percentual GC contents.

A corrected number of reads was estimated for each bin:

$$r_c = r_u - (r_{loess} - e(r)) \tag{2.5}$$

where r_u is uncorrected number of reads mapped to the given bin, r_{loess} is mean of read count mapped to bins with the same percentual GC content as GC content of the given bin, smoothed by the weighted local regression. $e(r)$ is expected average number of reads per bin for the chromosome, to which

the given bin belongs, and is determined as overall number of reads on the chromosome over the number of bins of selected size in the chromosome.

As in previous method, a sum of corrected read counts was computed for each chromosome and ratio of chromosome 21 was estimated using equation 2.1. A set of training samples were used to form normal distribution and z-score for each testing sample was determined with equation 2.2.

2.5 Within-sample GC correction

Sehnert et al [14] developed method for overcoming interrun sequencing variations. They propose to normalize the read counts on one chromosome with the total number of reads on predefined set of chromosomes. Presence of chromosomal mutation is determined using set of control samples. A chromosome ratio is obtained similarly as in equation 2.1, but sum of reads mapped to all autosomes is replaced by sum of reads of predefined chromosomes.

Straver et al [15] focused on abnormalities of this method trying to improve it. They pointed out, that different read frequencies are characteristic for different genome regions. Between-sample comparisons suffer from this. Within-sample comparisons at the other hand do not: Areas with equal read characteristics will behave the same way within a tested sample. They proposed a within-sample comparison using reference bins with similar behavior as a tested bin. They applied Loess normalization of GC content first and then used a set of healthy diploid samples to determine the within-sample reference bins with similar behavior as tested bin.

We implemented similar within-sample GC correction. First we identified a set of reference bins for each testing bin. The same set of training samples as in part 2.3 was used. Genome was divided into two parts: target - chromosome 21 (chromosome of interest) and reference - other autosomes (sex chromosomes were omitted in this method). The read pairs were filtered according to the method described in section 2.2. For each bin in both target and reference, number of reads mapped to this bin was determined for each

training sample.

In one of implementations Loess GC correction described in section 2.3 was applied to normalize read counts of target and reference part of sample. The other implementation performed only within-sample GC correction without previous use of Loess correction.

For each target bin squared Euclidean distance to all combinations of a reference bin and the target bin was computed using following equation:

$$D(i,j)^2 = \sum_{s=1}^{n} (R_{i_s} - R_{j_s})^2 \tag{2.6}$$

where $D(i,j)^2$ is squared Euclidean distance between target bin i and reference bin j, s iterates trough set of training samples, R_{i_s} and R_{j_s} are numbers of reads in sample s on target bin i and reference bin j, respectively. For each target bin a set of 100 reference bins with lowest squared Euclidean distance to the bin was selected.

Testing samples were divided into target bins from chromosome 21 (chromosome of interest) and reference bins from all other autosomes. Number of reads mapped to each bin was determined according to the same criteria that were used for training samples. Scores for each pair of target bin and reference bin were estimated, which allows us to evaluate presence of various chromosomal mutations (e.g. only a part of chromosome is redundant). However, we focused mainly on determination of final z-score for each sample. For obtaining final z-scores we used 3 different methods in order to see which method yields the most accurate results. We executed each of following methods with different bin sizes for both Loess corrected and uncorrected read counts.

2.5.1 Method 1

For each target sample and each reference bin (from the set of reference bins obtained from training samples) a z-score is estimated: Differences between number of reads on the specific reference bin and number of reads on all

other 99 reference bins is computed:

$$d_{jk} = r_j - r_k \tag{2.7}$$

where d_{jk} is difference between reference bins j and k, r_j and r_k are numbers of reads mapped to bin j and bin k in the testing sample, respectively. These 99 differences from reference bin j are expected to follow normal distribution $N(\mu_j, \sigma_j^2)$. We determine the difference d_t between target bin and reference bin j:

$$d_t = r_t - r_j \tag{2.8}$$

where r_t and r_j are numbers of reads on target bin and reference bin j in testing sample, respectively. A z-score is determined with following equation:

$$z = \frac{d_t - \mu_j}{\sigma_j} \tag{2.9}$$

where μ_j is mean and σ_j is standard deviation of normal distribution $N(\mu_j, \sigma_j^2)$ of differences d_{jk}.

That way 100 z-scores are obtained for each target bin. A score of a specific bin is defined as median of z-scores for this bin and a score for sample is defined as median of scores of its bins.

Scores of all training samples are obtained using this method, forming a normal distribution $N_1(\mu_1, \sigma_1^2)$. Final z-score of a testing sample is obtained with following equation:

$$z_{final} = \frac{z - \mu_1}{\sigma_1} \tag{2.10}$$

The sample was evaluated as trisomic if $z_{final} > 3$ and non-trisomic otherwise.

2.5.2 Method 2

The second method is very similar to Method 1: 100 z-scores for each target bin are obtained the same way as in 2.4.1 Method 1. This way we get a matrix Z of z-scores of size 100 x number of bins. The only difference between Method 1 and Method 2 is the way in which the score for sample is

defined. In Method 2 score for the sample is median of all values in matrix Z.

Scores of all training samples are obtained using this method, forming a normal distribution $N(\mu_2, \sigma_2^2)$. Final z-score of a testing sample is obtained with equation 2.10 using mean and standard deviation of distribution $N(\mu_2, \sigma_2^2)$. The sample was evaluated as trisomic if $z_{final} > 3$ and non-trisomic otherwise.

2.5.3 Method 3

A testing sample is divided into target and reference bins and number of reads mapped to each bin is estimated according to the same principles as in previous methods.

Relative read counts of the set of refference bins of each target bin follow normal distribution $N(\mu, \sigma^2)$. Score for target bin j is computed using following equation:

$$z = \frac{r_t - \mu}{\sigma} \tag{2.11}$$

where r_t is number of reads mapped to target bin, μ is mean and σ is standard deviation of normal distribution $N(\mu, \sigma^2)$. Score of a sample is defined as median of scores of bins.

Scores of all training samples are obtained using this method, forming a normal distribution $N_1(\mu_1, \sigma_1^2)$. Final z-score of a testing sample is obtained with equation 2.10. The sample was evaluated as trisomic if $z_{final} > 3$ and non-trisomic otherwise.

Chapter 3

Implementation

In this chapter we describe implementation of suggested methods. First we describe files used in our implementation and introduce used programming language and environment. Then we describe classes, that we created in our implementation.

3.1 Reference genome

For implementation and testing of our methods we used human reference genome (hg19), downloaded from the UCSC Genome site http://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/, that was produced in 2009 by Genome Reference Consortium.

Beginning of a chromosome in hg19.fa is denoted by line starting with '>' and containing name of the chromosome. Lines with bases of this chromosome follow until te end of chromosome. After that a line starting with '>' and denoting the beginning of next chromosome follows.

3.2 Sequenced samples

We used 110 different samples of sequenced genomes. All of the samples were tested for trisomies before and we were familiar with results of this testing.

Trisomy 21 occurred in 31 samples, rest of the samples were unaffected by any chromosomal mutations (diploid). Trisomies 13 and 18 were not present in our testing set. 60 diploid samples were used for implementation purposes in correction methods, allowing us to train characteristic behaviour of diploid samples. 31 samples with trisomy 21 and 19 diploid samples were used to test our methods of GC correction. We had a file in SAM format representing each of 110 samples.

3.3 SAM file format

SAM file format (Sequence/Alignment Map) specifies a structure for storing reads obtained from sequencing a DNA sample, uniquely mapped to the reference. SAM format supports single-end, pair-end reads and their combinations. Each SAM file consists of optional header section and mandatory alignment section. Data are stored in TSV (Tab Separated Values) format - a line symbolizes an read alignment and values of properties of each read alignment are separated by a tab stop character. An alignment line contains information about one read (in case of read pairs, two consecutive lines hold information about the pair). Each line has 11 mandatory and other optional fields. Mandatory fields must be present, but they can be filled with '*' or 0, depending on type of the field. [8]

Among others, each alignment line has field RNAME containing name of base sequence (chromosome), field POS holding the position to which the beginning of read was mapped, field MAPQ storing mapping quality, with which the read was mapped to reference and field TLEN storing the length of fragment. In case of read pairs it is length of fragment defined by the beginning of the first read and the end of the second read. [13] SAM file, that we used contained pair-end reads.

3.4 Java and Eclipse

We decided to implement proposed methods in Java language. Java is currently one of the most frequently used programming languages, offering rich user interface, portability and performance. Other major advantage of Java is that most of the classes and methods available are well documented.

The best method of GC correction should be used in application, that is already implemented in Java. Compatibility with this application was a main reason for using Java for our implementation.

For developement we used environment of Eclipse. Eclipse is a free and open-source integrated developement environment. It consists of workspace and extensible set of plug-ins that allow to accomodate the environment.

3.4.1 SAM Tools library

SAM Tools is a java package intended for work with files in SAM format. It provides various utilities for manipulating alignments in the SAM format, including sorting, merging and indexing. For work with SAM files we used classes SAMFileReader, SAMRecordIterator and SAMRecord from this package.

3.4.2 Commons Math library

Commons Math is the Apache Commons mathematics library of lightweight, self-contained mathematics and statistics components addressing the most common problems not available in the Java programming language or Commons Lang. For implementation of our classes we used classes LoessInterpolator and StandardDeviation.

3.5 Maven

Apache Maven is a software project management tool based on the concept of a project object model (POM). Maven can manage a project's build, reporting and documentation from a central piece of information. It provides a uniform build system allowing a project to build using its project object model (POM) and a set of plugins that are shared by all projects using Maven. One of the features of Maven is dependency management, which enables centralization of dependency information.

We used classes from SAM Tools and Common Math libraries. To integrate the libraries to our project we defined a dependencies in pom.xml file of the project. Following dependency was defined for SAM Tools library:

```
<dependency>
   <groupId>org.utgenome.thirdparty</groupId>
   <artifactId>picard</artifactId>
   <version>1.102.0</version>
</dependency>
```

3.6 Java classes

In this section we describe Java classes, that we implemented. Classes BinContent, GCContents, GcContentManager, SamFileChContents, ChrGcPercentage and CorrectedReadCounts provide methods for Loess GC correction. Class MainStatistics produces corrected and uncorrected read counts and class TrainingDistribution computes z-scores for testing samples using Loess corrected and uncorrected reads. Classes CorrelationStruct and BinCorrelation implement 3 methods of within-sample correction. Figures 3.1 to 3.10 show each of the implemented clases described in UML language.

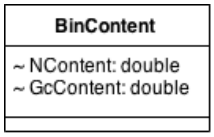

Figure 3.1: Class BinContent

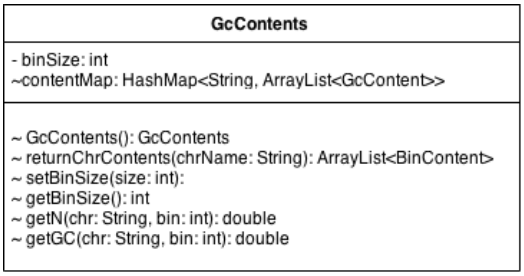

Figure 3.2: Class GcContents

3.6.1 BinContent

Class BinContent represents one bin of human reference genome (hg19) of arbitrary size and stores GC content of this bin. It has variables *NContent* and *GcCcontent*. *NContent* holds percentage of indeterminable bases in bin (denoted by N in hg19) and *GcContent* holds percentage of G and C bases in the bin. Percentage of A and T bases does not have to be stored since it can be computed as AtContent = 100 – *GcContent* – *NContent*.

3.6.2 GcContents

GcContents is a structure for storing GC content of the reference genome. It holds an information about GC content for each segment of specific size, that is stored in variable *binSize*. The actual bin size is set via class GC-ContentManager once the GC content of a sample is succesfully loaded.

GcContents also has a variable *contentmap* storing the actual GC content information of a sample. *contentmap* is of type HashMap<String, ArrayList <BinContent>>, where key is name of chromosome and value is an

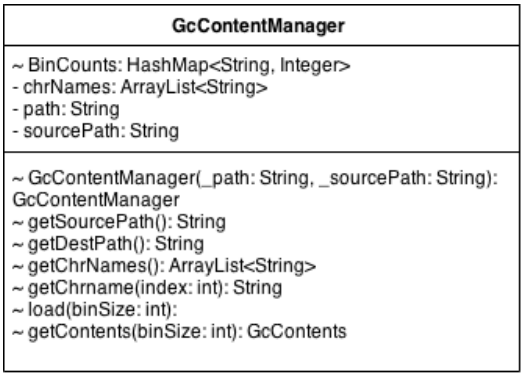

Figure 3.3: Class GcContentManager

ArrayList<BinContent> representing that chromosome.

GcContents has mainly get-methods returning bin size, ArrayList representing specific chromosome or GC and N content of specific bin on specific chromosome.

3.6.3 GcContentManager

GcContentManager is a class, that reads the file storing bases of reference human genome (hg19.fa) and computes GC content for fragments of desired size.

Method load() with parameter bin size loads hg19.fa, when a beginning of a chromosome is detected, the method creates a new .bin file, where numbers of GC and N bases in each bin of selected size are written. A .bin file storing such information has smaller size and is faster to process than a .txt file with the same content.

Method getContents() with argument bin size creates and returns GcContents structure, that is filled by data stored in .bin files from method load().

This class also stores a HashMap<String, Integer>, where key is a name of a chromosome and value is number of bins of this chromosome (corresponding to do bin size set in getContents() method).

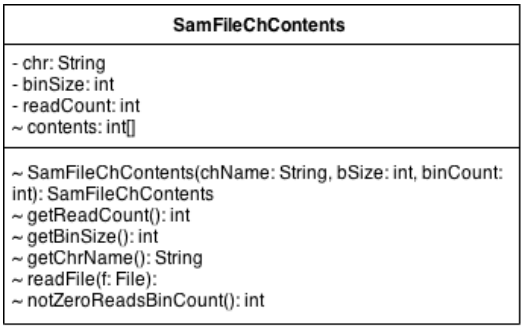

Figure 3.4: Class SamFileChContents

3.6.4 SamFileChContents

SamFileChContents is a class representing reads mapped to one chromosome.

The class has a method readFile() with argument file, which loads a specified SAM file and counts number of read pairs for each bin of the chromosome. The SAM file is loaded via SAMFileReader class from samtools package. SAMRecordIterator (class from samtools package) is obtained from SAMFileReader. Method next() of SAMRecordIterator returns SAMRecords. SAMRecord is a class from samtools package, that stores information about a single read or about one of reads in a read pair. The readFile() method iterates through SAMRecords and checks if each two consecutive reads (forming one read pair) fulfill criteria to be considered as correctly mapped and eligible for further analysis. A read pair was considered if both reads were mapped to the same chromosome (SAMRecord.getReferenceName()) with mapping quality ≥ 40 (SAMRecord.getMappingQuality()) and if the size of fragment defined by read pair was < 10000 (SAMRecord.getInferredInsertSize()). A read pair was mapped to the bin in which the first read of the pair was starting (SAMRecord.getAlignmentStart()). Number of reads mapped to each chromosome was stored in array of integers, where index stands for number of bin and value for number of read pairs mapped to this bin.

SamFileChContents also has a notZeroReadsBinCount(), that returns

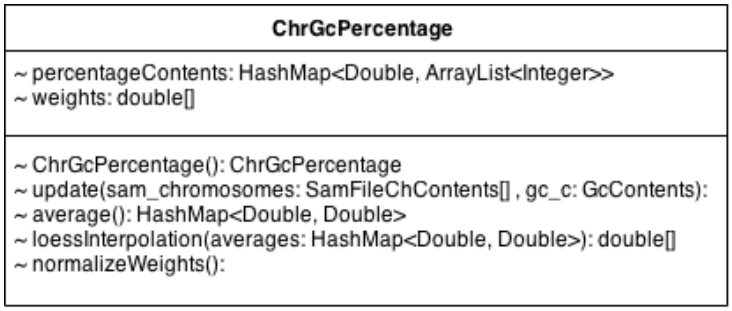

Figure 3.5: Class ChrGcPercentage

number of bins, to which more than 0 reads were mapped. This class also stores name of the chromosome and the bin size for which the mapping is executed.

3.6.5 ChrGcPercentage

This class contains Hashmap<Double, ArrayList<Integer>>, with GC content rounded to one decimal place as a key and ArrayList storing numbers of reads mapped to bin with corresponding GC content is value mapped to the key.

A method update(), with an array of SamFileChContents and a GcContents passed as arguments, loads number of reads of each bin to the ArrayList of corresponding GC content in the HashMap. An invariant is set: bin size stored in SamFileChContents and GcContents has to be equal. A bin is considered only if no N (undeterminable) bases occur in this bin in hg19.fa (Reads mapped to bins containing undeterminable bases are discarded).

A method average() computes average read count for bins with each possible GC content rounded to one decimal place and weight for each content (number of reads on the content over number of all reads)

Method loessInterpolation() executes weighted local regression on list of averages. For this purpose class LoessInterpolator from apache.amth.commons is used. Only GC contents with weight > 0 are used for interpolation.

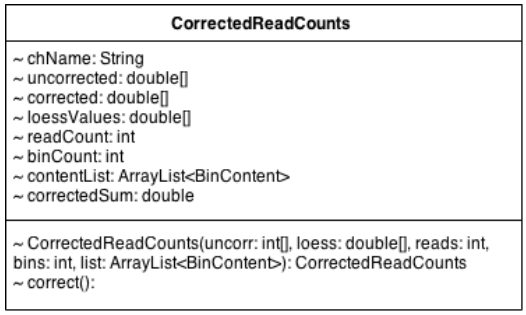

Figure 3.6: Class CorrectedReadCounts

3.6.6 CorrectedReadCounts

This class has only one method correct(), that corrects number of reads on a chromosome obtained from SAM file. If the original number of reads on bin was 0, the method will produce 0 as corrected value for this bin. Otherwise number of reads will be corrected:

```
double expected_val = (double)this.readCount/this.binCount;
for (int i = 0; i < uncorrected.length; i++){
   if (uncorrected[i] == 0){
      corrected[i] = 0;
   } else {
      BinContent c = contentList.get(i);
      double gcBinPercentage = c.GcContent;
      int index = (int)(gcBinPercentage*10);
      corrected[i] = uncorrected[i] -
      (this.loessValues[index] - expected_val);
   }
}
```

readCount and *binCount* are number of reads in the whole chromosome and number of bins of specified size in the chromosome, respectively. *uncorrected* is an array of uncorrected read values, *corrected* is an array

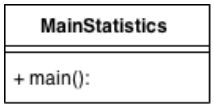

Figure 3.7: Class MainStatistics

of corrected read values. *contentList* is an ArraList storing information about GC content (in class BinContent) for each bin of the chromosome. $loessValues$ are smoothed mean values of number of reads for GC content (return value of method loessInterpolation() of class ChrGcPercentage). Method correct() also counts overall number of corrected reads on the chromosome.

3.6.7 MainStatistics

This is a class with a main method, in which the Loess correction is executed. The method is intended to be executed from command line with 4 arguments: reference to SAM file name, bin size, address of output directory and reference to hg19.fa.

GcContentManager and GcContents are created, number of reads is loaded for each chromosome via SamFileChContents class. Output .txt files with uncorrected numbers of reads are produced for each chromosome. ChrGcPercentage class is created, mean read numbers of reads for GC content are estimated and smoothed by loess interpolation. Object of class CorrectedReadCounts is created for each chromosome and read counts are corrected. Corrected values of read counts for each bin are written to .txt files (one file for one chromosome).

3.6.8 TrainingDistribution

Class TrainingDistribution has a main method, that loads .txt files with number of reads for each bin, chromosome and training sample (files produced by main method of class MainStatistics). Ratio of chromosome 21 is determined

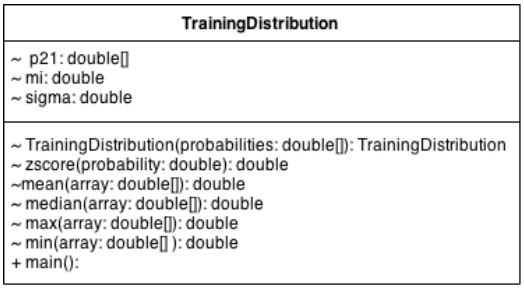

Figure 3.8: Class TrainingDistribution

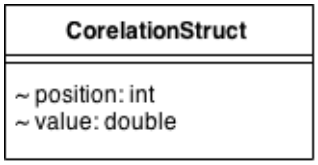

Figure 3.9: Class CorrelationStruct

for each of 60 training samples as sum of reads on chromosome 21 over sum of reads of all autosomes (chr1 – chr22). These ratios are expected to follow normal distribution. Mean and standard deviation of the distribution are obtained. Ratio of chromosome 21 is estimated for each sample and z-score is assigned:

```
double z_score = (probability - this.mi)/this.sigma;
```

where mi and sigma are mean and standard deviation of the distribution, respectively. The main method executes this computations first with uncorrected numbers of reads and then with corrected numbers of reads (computed by class MainStatistics). The resulting z-scores are written to a .txt file.

3.6.9 CorrelationStruct

This class works as a structure for storing one integer – *position*, and one double – *value*. CorrelationStruct represents correlating reference bin of a

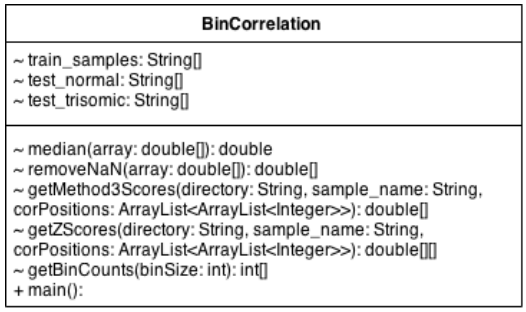

Figure 3.10: Class BinCorrelation

target bin in within-sample GC correction. *position* stores the position of reference bin and *value* stores value of squared Euclidean distance between corresponding reference bin and a target bin.

3.6.10 BinCorrelation

This class has a method removeNaN() which takes array of doubles as an argument and returns new array without NaN values.

It has a main method in which ArrayList<ArrayList<Double>> is created to store number of reads on each bin of each training sample. ArrayList <Double> is a list of read counts for one bin, where read count for different sample is on each index of the list (there are 60 training samples, hence 60 read counts in one list). ArrayList<ArrayList<Double>> then represents one or more chromosomes. Read counts are loaded from .txt files produced by main method of class MainStatistics.

An empty list of correlating reference bins is created for each sample. Relative numbers of reads are determined – a read count on specific bin and sample is divided by number of all reads of that sample. Squared Euclidean distance is computed for each target bin and each reference bin:

```
double SqEuclidianDist = 0;
for (int k = 0; k < correlation.train_samples.length; k++){
```

```
    double rel21 = b21.get(k)/overallReads.get(k);
    double relO = bOther.get(k)/overallReads.get(k);
    SqEuclidianDist += (rel21 - relO) * (rel21 - relO);
}
```

where $b21$ represents target bin and $bOther$ represents reference bin. k iterates through the set of training samples. If size of list of correlating reference bins for current target bin is lower than 100, the reference bin is added to the list. Otherwise a bin with maximal Euclidean distance in the list is determined and its value is compared to the Euclidean distance of the currently processed reference bin. If the Euclidean distance of currently processed reference bin is lower, it replaces the bin with maximal value in the list. This way a 100 correlating reference bins with lowest Euclidean distance are determined for each target bin.

For each target bin a sum of squared Euclidean distances of correlating reference bins is checked. If the sum is 0, the target bin is most probably one of the bins, that contain indeterminable bases and therefore no reads were mapped to this bin. We do not want to consider such bins in further analysis, so list of correlating reference bins is in such case filled with -1.

100 z-scores are determined for each target bin in method getZscores(). If the list of correlating bins of a target bin is filled with -1, NaN values are returned as z-scores for that bin. Otherwise a list of differences is created. Each of them stands for difference from current correlating reference bin to other of the correlating reference bins, which is together 99 differences for each correlating reference bin of each target bin. These 99 values follow normal distribution, of which mean and standard deviation are obtained. For obtaining a standard deviation class StandardDeviation from package apache.math.commons is used. A difference from target bin to current correlating reference bin is determined and z-score for the pair (current target bin and current reference bin) is estimated:

```
double r21 = reads21.get(i)/overallReads;
```

```
double diff = r21 - corelating[j];
z_scores[i][j] = (diff - mean)/deviation;
```

After that a score is determined for each sample twice: first a bin median value is established from 100 z-scores of each target bin and a score of the sample is determined as median of bin medians. Secondly a sample score is determined as median of all the values in the matrix of z-scores. In order to obtain a median value NaN values are removed from the corresponding array by method removeNaN(), because Java considers NaN as an actual number with high value, which produces higher median.

Scores for training samples are obtained the same way, forming a normal distribution. A mean and standard deviation of the distribution are estimated. A final z-score of each testing sample is determined.

Final z-scores according to method described in section 2.4.3 are obtained for testing samples in method getMethod3Scores(). Corrected and uncorrected reads are loaded from .txt files produced by main method of class MainStatistics. For each target bin a list of correlating reference bins is present. If the list of correlating bins is filled with -1, NaN is set as a z-score of that target bin. Otherwise relative read counts on correlating reference bins are determined, forming a normal distribution. Mean and standard deviation of the distribution are estimated and a bin score is determined:

```
double sum = 0;
for (int j = 0; j < cor_notNaN.length; j++){
   sum += cor_notNaN[j];
}
double mean = sum/(double)cor_notNaN.length;
StandardDeviation std = new StandardDeviation();
double dev = std.evaluate(cor_notNaN);
double r21 = reads21.get(i)/overallReads;
scores[i] = (r21 - mean)/dev;
```

Score of a sample is obtained as median of bin scores.

Scores of training samples and testing samples are obtained. Scores of training samples form a normal distribution. Mean and standard deviation of this distribution ar determined and a final z-score for each testing sample is obtained.

Final z-scores obtained from all methods are written to .txt files.

Chapter 4

Results

4.1 Sequenced samples

We used 110 samples of sequenced DNA, read alignments of each sample were described in separate SAM file. All of the samples were previously tested for aneuploidies with invasive diagnostic methods and we were familiar with outcome of the testing: out of 110 samples 31 were trisomic (trisomy 21), all other (79) samples were diploid. Trisomy 13, 18, trisomies of sexual chromosomes or other aneuploidies were not present in our set of samples. That is why we examined only presence of trisomy 21 in all samples.

From 110 samples 60 diploid samples were used as a training set for the proposed methods. Other 50 samples were used for testing. The testing set consists of 31 trisomic and 19 diploid samples. By performing the tests we intended to obtain confirmation of previous diagnosis for the samples.

4.2 Testing

At first, results of a regular prediction of trisomy 21 without any correction (section 2.2) were obtained. Secondly, the Loess correction (section 2.3) was applied, corrected read counts and z-scores of this method were obtained.

Consequently all within-sample correction methods (section 2.4) were applied to uncorrected as well as to corrected read counts and z-scores from all methods were determined.

In all the methods, that we implemented, an arbitrary bin size could be selected. All of the methods were performed with 3 different bin sizes: 20 000, 100 000 and 500 000 bases. Figures 4.1, 4.2 and 4.3 illustrate z-scores of testing samples for different methods with bin size 20 000, 100 000 and 500 000, respectively. On x-axis UN stands for uncorrected and CO for corrected set of read counts, the first number describes bin size (in 1000 bases) and the second number denotes the method used. The resulting z-scores of each method are presented in Table 4.1 for non-trisomic and Table 4.2 for trisomic samples. For each trisomic and non-trisomic set of samples a minimum, median and maximum of obtained z-scores are shown.

In some GC correction methods (Method 1, Method 2) applied to bins of size 20 000, the maximal non-trisomic z-score rose over 3 when the Loess correction was omitted and to over 2.5 when the Loess correction was applied first. The version of Method 3 without Loess correction applied to bin size of 20 000 bases produced minimal trisomic z-score only slightly higher than 3. Desired outcome would be if there were no samples with z-scores from interval $[2.5; 4]$ as such results might be caused by some imprecision in testing and therefore are not reliable enough. Diploid samples should have z-scores lower than 2.5, trisomic samples should have z-scores greater than 4. Even though Methods 1 and 2 produce higher non-trisomic z-scores when applied to bins of lower sizes, the trisomic z-scores are considerably higher than 3 and therefore the difference line between trisomic and non-trisomic samples can still be clearly established. The same way Method 3 applied to bins of lower sizes produces lower trisomic z-scores, but considerably lower non-trisomic z-scores at the same time. The difference line between non-trisomic and trisomic samples can be clearly established also in this case.

In all cases application of Loess correction improved the results - median trisomic z-score determined by method with Loess GC correction was always

higher than a median trisomic z-score obtained by the same method without the use of Loess GC correction. Only in within-sample correction Method 2 for bins of size 500 000, the minimal and maximal z-score decreased with use of Loess correction, however this descent is not significant and the median z-score is still higher with use of Loess correction. On average the Loess correction helped to increase the difference between the maximal non-trisomic and minimal trisomic z-score.

The most successful method of GC bias correction was Loess correction combined with Method 2 of within-sample correction applied to bins of size 20 000 bases. The highest rise in trisomic z-scores was observed after applying this method: the median z-score rose by almost 5 and the maximum z-score rose by almost 11. Method 2 combined with Loess correction generally performed well in combination with all tested bin sizes.

In respect to correction of both trisomic and non-trisomic z-scores, was Loess correction combined with Method 3 of within-sample correction most successful, as the z-scores obtained by this method never lay in interval $[2.5; 4]$.

In case of within-sample correction methods, the non-trisomic median z-scores were expected to be close to 0. In order to check the accuracy of proposed within-sample correction methods, the z-scores of non-trisomic training samples were determined and processed. The median z-scores of training samples with use of different methods and bin sizes are shown in Table 4.3. Median z-scores of training samples are close to 0, which shows that within-sample correction methods were implemented correctly.

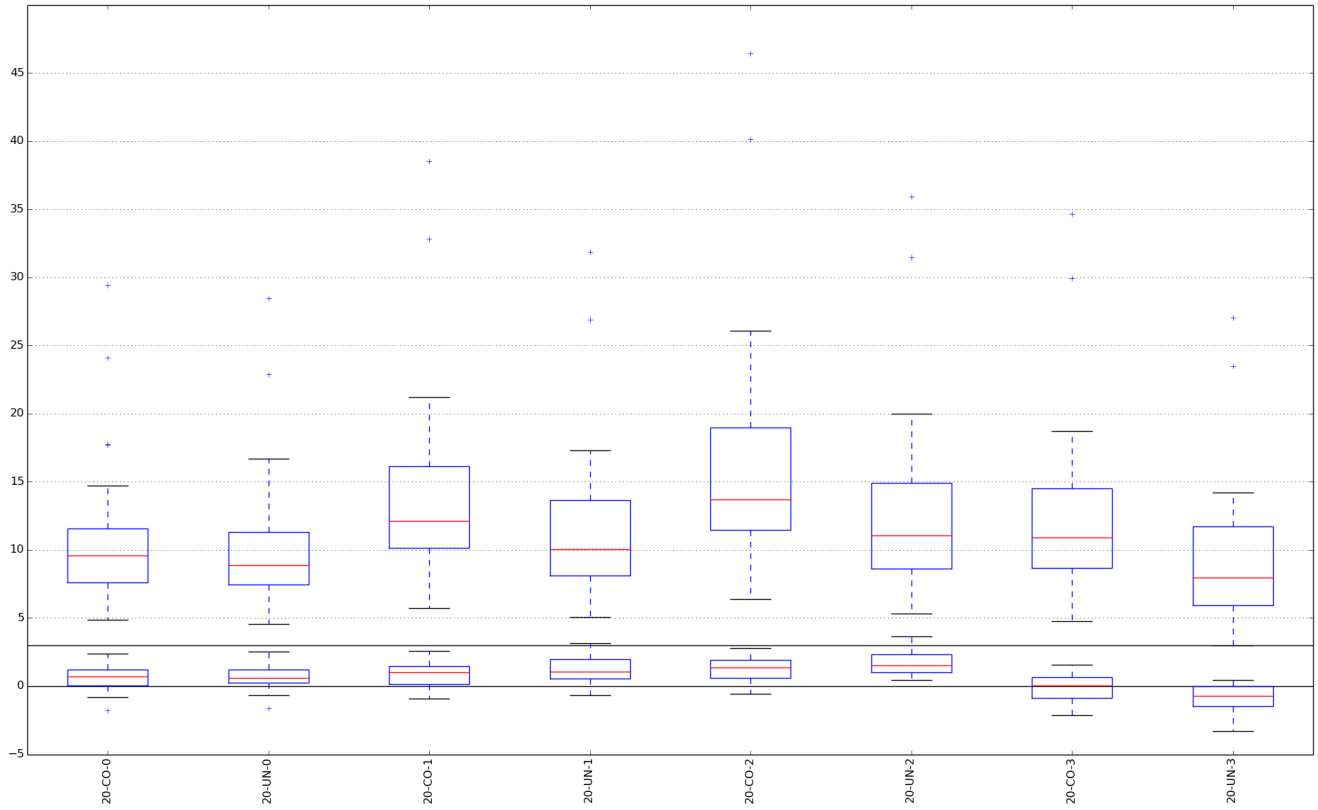

Figure 4.1: Results for bin size 20 000

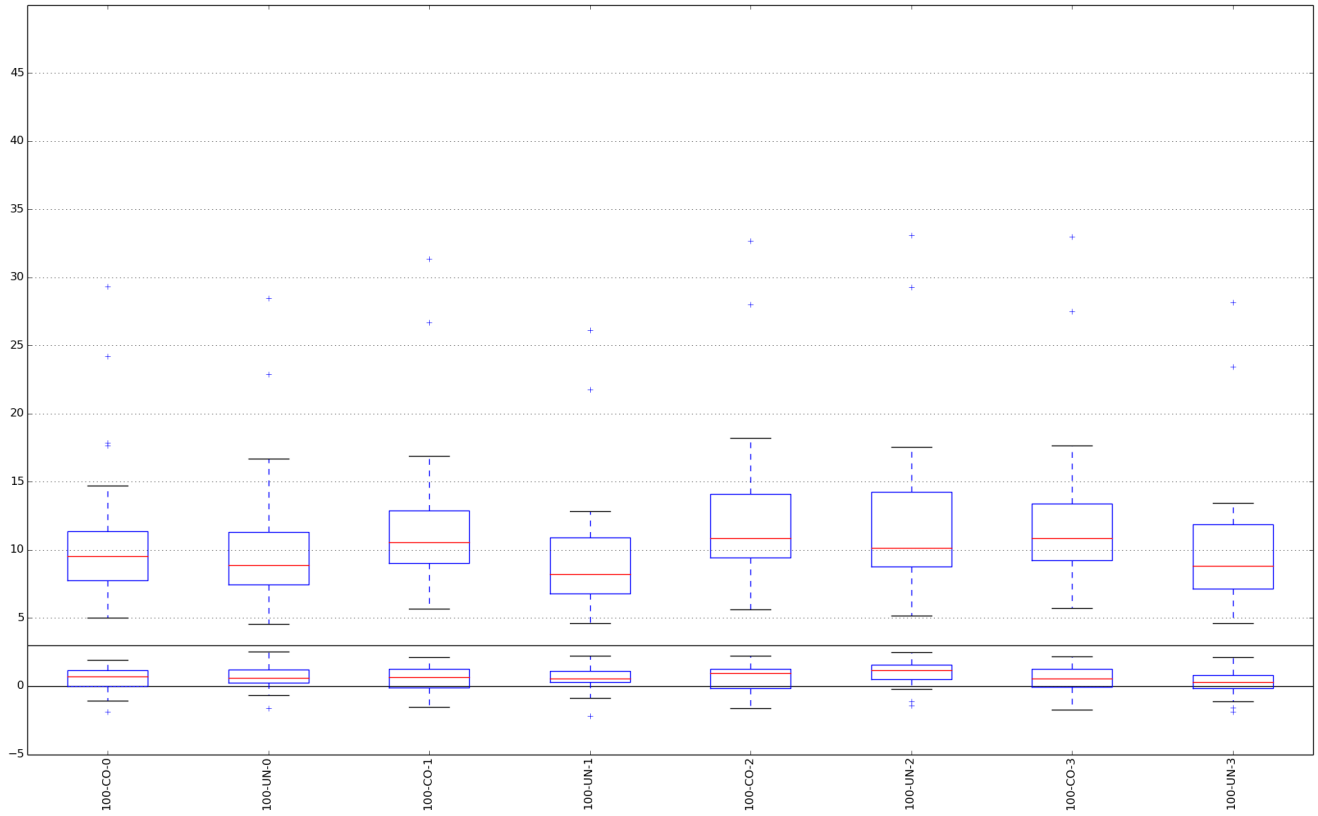

Figure 4.2: Results for bin size 100 000

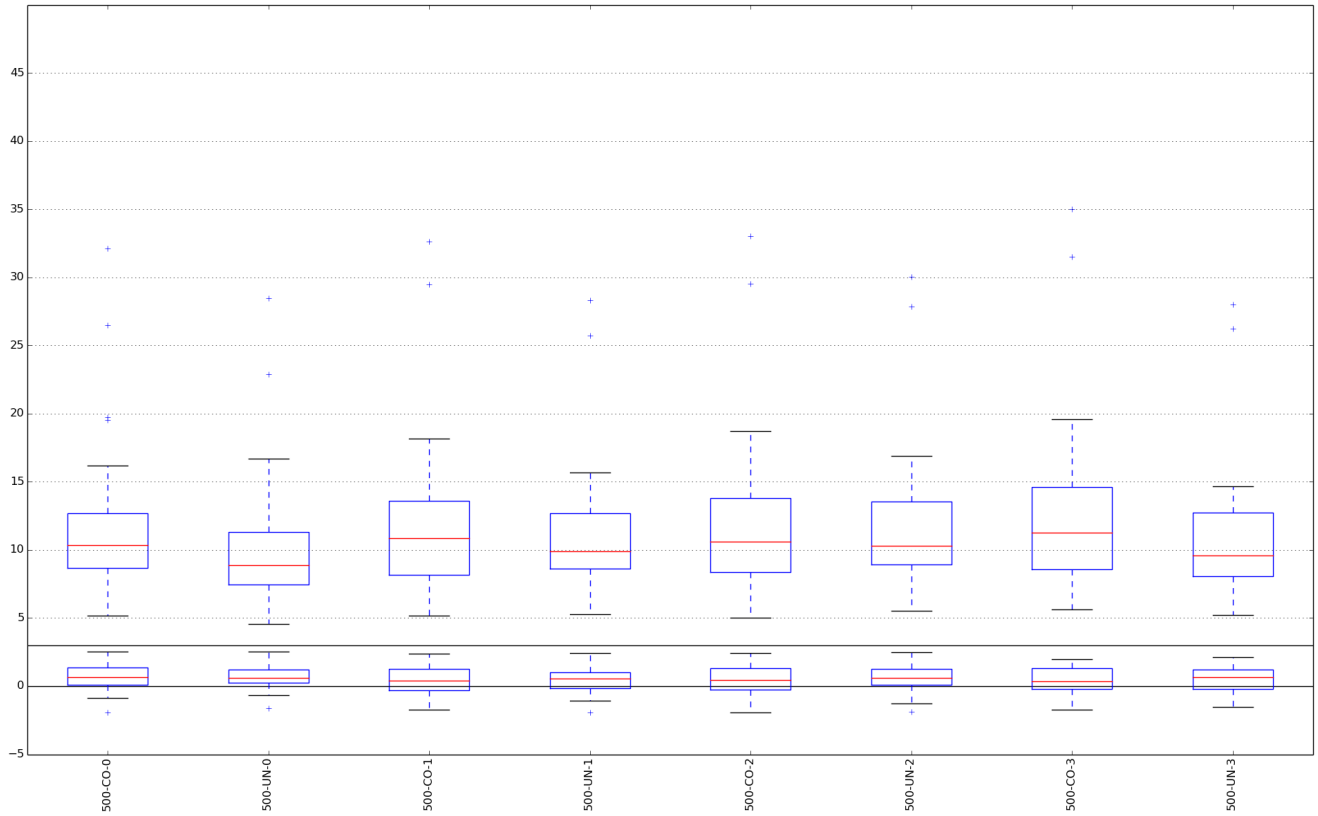

Figure 4.3: Results for bin size 500 000

Table 4.1: z-scores of trisomic samples

Bin size	Method	Minimum	Median	Maximum
-	without correction	4,57	8,88	28,46
20000	Loess correction	4,87	9,56	29,45
	Method 1	5,05	10,03	31,85
	Loess + Method 1	5,74	12,11	38,52
	Method 2	5,30	11,07	35,92
	Loess + Method 2	**6,38**	**13,69**	**46,43**
	Method 3	3,01	7,98	27,04
	Loess + Method 3	4,76	10,90	34,65
100000	Loess correction	5,00	9,54	29,32
	Method 1	4,59	8,21	26,11
	Loess + Method 1	5,66	10,56	31,34
	Method 2	5,18	10,14	33,06
	Loess + Method 2	5,62	10,87	32,67
	Method 3	4,63	8,84	28,16
	Loess + Method 3	5,74	10,84	32,98
500000	Loess correction	5,00	9,54	29,32
	Method 1	5,29	9,90	28,33
	Loess + Method 1	5,66	10,56	31,34
	Method 2	5,18	10,14	33,06
	Loess + Method 2	5,01	10,61	33,02
	Method 3	5,24	9,57	27,98
	Loess + Method 3	5,64	11,24	35,02

Table 4.2: z-scores of non-trisomic samples

Bin size	Method	Minimum	Median	Maximum
-	without correction	-1,63	0,62	2,52
20000	Loess correction	-1,79	0,69	2,40
	Method 1	-0,67	1,06	3,14
	Loess + Method 1	-0,89	1,00	2,56
	Method 2	0,46	1,50	3,64
	Loess + Method 2	-0,57	1,36	2,78
	Method 3	-3,31	-0,73	0,44
	Loess + Method 3	-2,15	0,03	1,55
100000	Loess correction	-1,90	0,71	1,90
	Method 1	-2,21	0,58	2,25
	Loess + Method 1	-1,52	0,67	2,11
	Method 2	-1,40	1,15	2,48
	Loess + Method 2	-1,61	0,97	2,21
	Method 3	-1,88	0,30	2,13
	Loess + Method 3	-1,14	0,60	2,19
500000	Loess correction	-1,90	0,71	1,90
	Method 1	-1,90	0,55	2,44
	Loess + Method 1	-1,52	0,67	2,11
	Method 2	-1,40	1,15	2,48
	Loess + Method 2	-1,91	0,48	2,45
	Method 3	-1,52	0,64	2,15
	Loess + Method 3	-1,72	0,36	1,99

Table 4.3: Median z-scores of non-trisomic training samples produced by application of within-sample correction methods

Bin size	GC correction method	Median z-score of training samples
20000	Method 1	0,01
	Loess correction + Method 1	-0,14
	Method 2	-0,06
	Loess correction + Method 2	-0,06
	Method 3	0,12
	Loess correction + Method 3	0,12
100000	Method 1	0,05
	Loess correction + Method 1	0,07
	Method 2	0,06
	Loess correction + Method 2	-0,03
	Method 3	-0,05
	Loess correction + Method 3	0,02
500000	Method 1	-0,09
	Loess correction + Method 1	-0,02
	Method 2	-0,03
	Loess correction + Method 2	0,01
	Method 3	0,04
	Loess correction + Method 3	-0,10

Conclusion

The main aim of our work was to propose GC correction method that would eliminate z-scores from interval $[2.5; 4]$, as those are not sufficiently reliable. We implemented general method of trisomy 21 prediction. Based on published articles the most common method of GC correction – Loess correction, was implemented. As a second approach, three different within-sample correction methods were implemented and run on raw data as well as on data corrected by Loess correction.

The testing on set of trisomic and non-trisomic samples showed, that Loess correction improved the quality of resulting z-scores in all the cases. Different correction samples influenced z-scores to various extent. In most cases the values of trisomic samples lay above 4 and of non-trisomic samples below 2.5. Method 3 of within-sample correction combined with Loess correction was observed to be the most efficient method in terms of yielding non-trisomic z-scores lower than 2.5 and trisomic z-scores higher than 4. Besides that Loess correction combined with Method 2 of within-sample correction with use of bins of size 20 000 was identified as the most efficient method in terms of increasing trisomic z-scores - the mean trisomic z-score increased by 54% in comparison with uncorrected data. The sharpest rise in trisomic z-scores was noted when bins of lower size (20 000) were used.

Within-sample correction methods offer a more precise view on chromosome ratios in various parts of genome. In this work we focused only on establishing a z-score for the whole sample. However determination of a z-score for each bin of the samples creates space for prediction of partial

chromosomal mutations (loss or gain of part of a chromosome).

Bibliography

[1] Yuval Benjamini and Terence P. Speed. Summarizing and correcting the gc content bias in high-throughput sequencing. Nucleic acids research, pages 1-14, 2012.

[2] Chen, Eric Z. et al. Noninvasive prenatal diagnosis of fetal trisomy 18 and trisomy 13 by maternal plasma dna sequencing. PloS one, 2011.

[3] William S. Cleveland and Susan J. Devlin. Locally Weighted Regression: An approach to regression analysis by local fitting. Journal of the American Statistical Association, 83(403):596-610, 1988.

[4] Gálová, Ševčovičová, Miklovičová, Švec (2004), Vybrané texty a príklady k cvičeniam z genetiky, Prírodovedecká fakulta Univerzity Komenského

[5] Griffiths AJF, Miller JH, Suzuki DT, et al. An Introduction to Genetic Analysis. 7th edition. New York: W. H. Freeman; 2000. Aneuploidy. Available from: http://www.ncbi.nlm.nih.gov/books/NBK21870/

[6] Illumina: An introduction to next generation sequencing

[7] Illumina, Quality scores for next-generation sequencing: Assessing seuencing accuracy using Phred quality scoring.

[8] Li H, Handsaker B, Wysoker A, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25(16):2078-2079. doi:10.1093/bioinformatics/btp352.

[9] Liao C, Yin AH, Peng CF, Fu F, Yang JX, et al. (2014) Noninvasive prenatal diagnosis of common aneuploidies by semiconductor sequencing. Proc Natl Acad Sci U S A 111: 7415–7420. doi: 10.1073/pnas.1321997111

[10] Lo YM, Corbetta N, Chamberlain PF, et al. Presence of fetal DNA in maternal plasma and serum. Lancet. 1997;350(9076):485–487

[11] Luthardt, Frederick W and Keitges, Elisabeth, Chromosomal syndromes and genetic disease, eLS, 2001, Wiley Online Library

[12] O'Connor, C. (2008) Chromosomal abnormalities: Aneuploidies. Nature Education 1(1):172

[13] The SAM/BAM Format Specification Working Group, Sequence Alignment/Map Format Specification, 11 May 2015, https://github.com/samtools/hts-specs.

[14] Sehnert A, Rhees B, Comstock D, deFeo E, Heilek G, Burke J, Rava R. Optimal detection of fetal chromosomal abnormalities by massively parallel DNA sequencing of cell-free fetal DNA from maternal blood. Clin. Chem. 2011;57:1042–1049

[15] Straver, Roy et al. "WISECONDOR: Detection of Fetal Aberrations from Shallow Sequencing Maternal Plasma Based on a within-Sample Comparison Scheme." Nucleic Acids Research 42.5 (2014): e31. PMC. Web. 22 May 2015.

[16] Zvelebil, Marketa J. and Baum, Jeremy O. (2008), Understanding Bioinformatics, Garland Science