VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

DIFFERENTIAL GENE EXPRESSION USING A NEGATIVE BINOMIAL MODEL

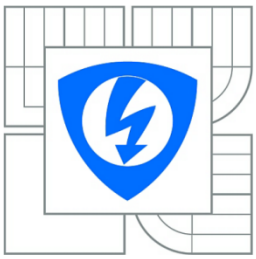
DIFERENCIÁLNÍ EXPRESE GENŮ NA ZÁKLADĚ NEGATIVNÍHO BINOMICKÉHO MODELU

DIPLOMOVÁ PRÁCE
MASTER THESIS

AUTOR PRÁCE — Bc. TEREZA JANÁKOVÁ
AUTHOR

VEDOUCÍ PRÁCE — Ing. LAYAL ABO KHAYAL
SUPERVISOR

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav biomedicínského inženýrství

Diplomová práce

magisterský navazující studijní obor
Biomedicínské inženýrství a bioinformatika

Studentka: Bc. Tereza Janáková ***ID:*** 125030
Ročník: 2 ***Akademický rok:*** 2013/2014

NÁZEV TÉMATU:

Diferenciální exprese genů na zakladě negativního binomického modelu

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši témat sekvenování RNA, Next Generation Sequencing (NGS) a Pherd quality scale. 2) Vyberte gen vhodný ke studiu, získejte short reads a vytvořte soubory ve formátu SAM s využitím mapovače Bowtie2. 3) Vytvořte anotované objekty genů s použitím složby Ensembl's BioMart a proveďte zarovnání dat. 4) Určete genovou expresi, odvoďte difereciální signály exprese RNA a odhadněte faktory velikosti knihovny. 5) Proveďte odhad parametrů negativního binomiálního rozložení. 6) Vyhodnoťte výsledky a proveďte diskusi. Projekt bude řešen a vypracován v anglickém jazyce.

DOPORUČENÁ LITERATURA:

[1] DI, Y., SCHAFER, D. W., CUMBIE, J. S., CHANG, J. H. The NBP Negative Binomial Model for Assessing Differential Gene Expression from RNA-Seq . Statistical Applications in Genetics and Molecular Biology, Volume 10, Issue 1, Pages 1–28, ISSN (Online) 1544-6115, 2011.
[2] SONESON, C., DELORENZI, M. A Comparison of Methods for Differential Expression Analysis of RNA-seq Data. BMC Bioinformatics; 14:91, 2013.

Termín zadání: 10.2.2014 ***Termín odevzdání:*** 23.5.2014

Vedoucí práce: Layal Abo Khayal
Konzultanti diplomové práce: prof. Ing. Ivo Provazník, Ph.D.

prof. Ing. Ivo Provazník, Ph.D.
Předseda oborové rady

UPOZORNĚNÍ:

Abstract

The main goal of this master thesis is to carry out the analysis of differential gene expression using a negative binomial model. The first part is devoted to theoretical basis, discusses the RNA sequencing, Next-Generation Sequencing (NGS), the benefits and applications, and FASTAQ format. The second part is the practical part, there was chosen a suitable data set of genes, that will be later analyzed, and the relevant data was downloaded. This data was aligned to the human genome version 37 by Burrows-Wheeler transform and the SAM formatted files were created using the *Bowtie* mapper. The SAM formatted files were sorted by SAMtools. In the following part of this work was created an annotation object of target genes using Ensembl´s BioMart service and Matlab (version R2013b). Next, digital gene expression was determined and library size factor was estimated. In the end the negative binomial distribution parameters were estimated and data was tested for a differential gene expression.

Keywords

RNA-sequencing, Next Generation Sequencing, Differential gene expression, Prostate Cancer

Abstrakt

Hlavním cílem této diplomové práce je analýza diferenciální exprese genů na základě negativního binomického modelu. Úvodní část je věnována teoretickému základu, pojednává o sekvenování RNA, sekvenování nové generace, výhodách a možném využití, formátu fastQ aj. Následující část už se zabývá samotnou praktickou částí, zde byl vybrán vhodný set genů, které budou později analyzovány a příslušná data byla stažena. Tato data byla zarovnána k lidskému genomu verze 37 Burrowsovou-Wheelerovou transformací s využitím *bowtie* mapovače, byly tak vytvořeny soubory ve formátu SAM. Toto soubory dat byly později setříděny pomocí nástroje SAMtools. Následně byly v programovém prostředí Matlab (verze R2013b) vytvořeny anotované objekty genů s využitím služby Ensembl´s BioMart. Dále byla určena genová exprese a byly odhadnuty faktory velikosti knihovny. Na závěr byly odhadnuty parametry negativního binomického rozložení a byla vyhodnocena diferenciální exprese genů.

Klíčová slova

RNA-sekvenování, sekvenování nové generace, diferenciální genová exprese, rakovina prostaty

Bibliografická citace mé práce:

Bc. JANÁKOVÁ, T. Differential gene expression using a negative binomial model. Brno: Brno University of Technology, Faculty of electrical engineering and communication, department of biomedical engineering, 2014. 67 pages. Supervisor of master thesis: Ing. Layal Abo Khayal

Declaration

I declare that I have elaborated my master thesis on the theme of "Differential gene expression using a negative binomial model" independently, under the supervision of the master's thesis supervisor and with the use of technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the master thesis I furthermore declare that, concerning the creation of this master's thesis, master's thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone's personal copyright and I am fully aware of the consequences in the case of breaking Regulation S 11 and the following of the Copyright Act No 121/2000 Vol., including the possible consequences of criminal law resulted from Regulation S 152 of Criminal Act No 140/1961 Vol.

Brno…… .....
author´s signature

Acknowledgment

I would like to thank to my supervisor, Ing. Layal Abo Khayal, for her help in achieving the goals and the writing of this thesis. Special thanks belongs to my semestral consultant, prof. Ing Ivo Provazník, Ph.D. for his patience, wise advises and support in the beginning of this thesis.
Lastly, my great love and appreciation to my parents and family for supporting my studies.

Brno…… .....
author´s signature

INTRODUCTION

Gene expression is the process by which information from a gene is used in the physiological synthesis of functional gene products; the most important are proteins. All somatic cells in our body contain the same genetic information, but each cell type expresses a unique subset of all encoded genes. This is because each type of cell expresses different genes, this mechanism is known as "differential gene expression." Every organism regulates its gene expression to achieve developmental changes, cellular specialization or adaptation to a new environment. Analysis of gene expression patterns provides a valuable understanding of the normal biological and disease processes. The most percentage of differential gene expression analysis is now focused on cancer diseases. The scientists are studying the differences between the gene expression of treated and non-treated cells. The research can bring huge possibilities in the future to provide an answer, how to affect part of the genome in a way that a diseased person becomes healthy again. [47]

Differential gene expression can be controlled at many levels. Transcription occurs at the first stage of the expression. RNA-sequencing, called also whole genome sequencing, is an emerging technology for surveying gene expression at the transcript resolution. The recent Next-Generation Sequencing (NGS) methods have been developed during the last decade. Those revolutionary technologies provide cheap, fast and correct information about the DNA sequencing. [9]

The aim of this work is identifying differentially expressed genes from RNA-Seq Data by a selected method implemented in MATLAB using its bioinformatics and statistics toolboxes. This thesis deals with various methods of Next-Generation Sequencing, their advantages and applications, and provides theoretical basis.

In the practical part, we choose a set of interested genes, as prostate cancer data set and download their sequences to analyze them by MATLAB environment. Following the NGS data analyzing workflow, we produce SAM formatted files by mapping the downloaded short reads to the whole human genome. For this task, we use *Bowtie* mapper under Linux as an implementation of a method to map short reads to a reference sequence using Burrows-Wheeler transform. Then we sort the mapped reads in SAM format using SAMtools. For the subsequent analysis we create an annotation object of the targeted genes. Then use Ensemle´s BoiMart service to download the table of all protein encoding genes and load this table in MATLAB to create an annotation file. To create a BioMap, which is an object enables better manipulation of the sequences, we import the mapped short reads into MATLAB. To

determine the digital gene expression, we use the previously created BioMap objects and the bioinformatics toolbox including its implemented functions. Then we study the inference of differential signal in RNA expression, this means estimating library size factor and negative binomial distribution parameters. Eventually we test the data for differential gene expression, to figure out how the gene expression was affected in DHT-treated sample.

1 BACKGROUND INFORMATION

1.1 Genomic sequencing

As the DNA carries the genetic information, we need to know the nucleotide order of DNA fragments. The process of getting this information is called sequencing. By sequencing we can find the epigenetic changes which are affected by the gene expression, for example methylation. DNA sequences are essential virtually for all branches of biological research. One of the first methods; Sanger sequencing has always been restricted by inherent limitations (throughput, scalability, speed, and resolution). To overcome these barriers, a new technology was required; Next-Generation Sequencing (NGS). In last 50 years the field of molecular biology made a huge progress, specially sequencing, in the beginning we could read only few bp (base pair), nowadays we are able to read millions of reads in only few hours.

Next-Generation Sequencing is called also second or third generation sequencing, deep or ultra-deep sequencing or massive parallel sequencing. With discovery of the NGS, the cost per Mb of DNA sequence has rapidly decreased. The same effect is seen on the cost per genome. (Figure 1, Figure 2) [1]

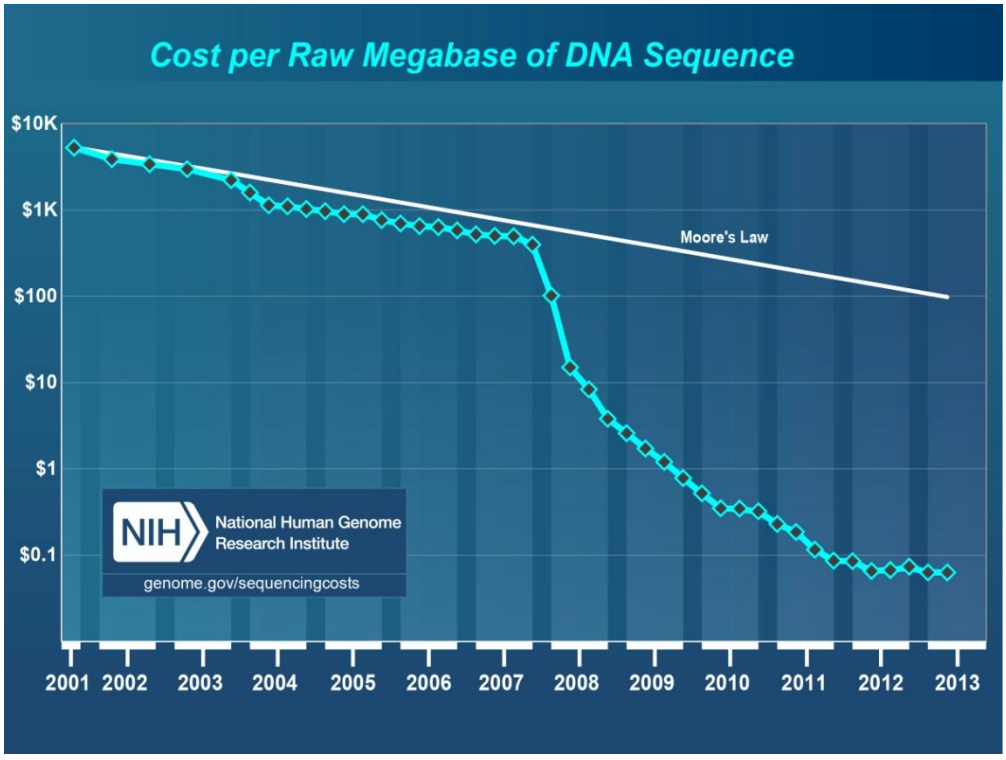


Figure 1: Cost per Megabase of DNA sequence has rapidly decreased (www.genome.gov)

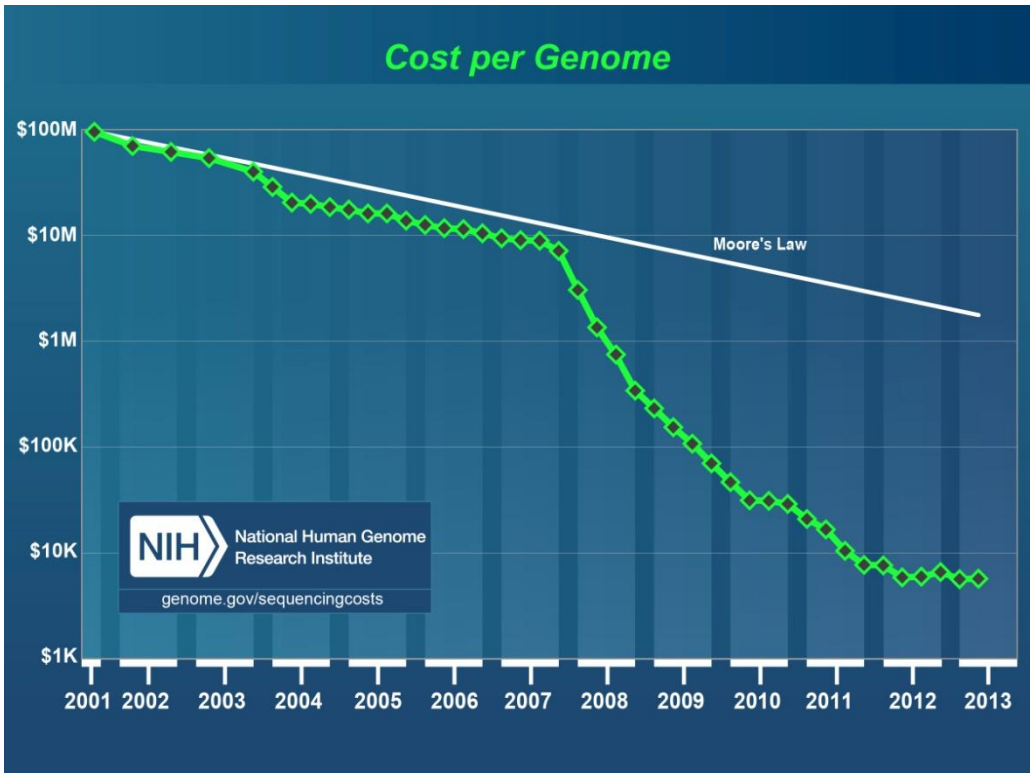


Figure 2: Cost per Human Genomes rapid decrease (www.genome.gov)

1.2 RNA sequencing – view the whole transcriptome

Transcriptome analysis is an important tool for characterizing and understanding of the molecular basis of phenotypic variation in biology, including diseases. In the previous time, DNA microarrays have been used to quantify of mRNA, which corresponds to different genes, but recently high-throughput sequencing of cDNA (RNA-seq) has come out as a powerful competitor. The use of RNA-seq for differential expression analysis will rapidly increase, because the cost of sequencing decreases.

RNA sequencing workflow, from sample preparation through data analysis, enables rapid profiling and deep investigation of the transcriptome. With the greatest daily output available for any sequencing system, transcript profiles can be viewed in a single day.

RNA sequencing reads can be aligned across splice junctions and isoforms, novel transcripts and gene fusion can be identified. [2]

1.2.1 Application of RNA-Seq

mRNA-Seq

mRNA-Seq delivers unbiased and unparalleled information about the transcripotome, that all with no probes or primers to design. "Stranded" information identifies from which of the two DNA strands was the given transcript delivered. It provides increased trust in transcript annotation particularly for non-human samples and may serve to increase the percentage of aligned reads, and reducing sequencing costs per sample. Strand orientation also provides the detection of antisense expression, providing visibility to regulatory relationships that would otherwise be missed. [3]

Total RNA-Seq

Whole-transcriptome analysis with total RNA-seq covers a wide range of gene expression changes and enables the detection of novel transcripts in both coding and non-coding RNA types. Ribo-Zero ribosomal RNA reduction chemistry, removes efficiently ribosomal RNA (rRNA), using a hybridization/bead capture procedure, that selectively binds target sequences using biotinylated capture probes. This process minimizes ribosomal contamination and optimizes the percentage of reads covering RNA species that are interesting. [3]

Paired-End RNA-Seq

Paired-End RNA-Seq is a universal application using 200-500 bp fragments, paired-end libraries [3]. It is strategy for genome-wide, high-resolution identification of fusion genes and other large scale rearrangements. It can be used for paired-end sequencing of clones, or other fragments of genomic DNA, from tumor samples. The resulting paired reads, are mapped back to the reference human genome sequence. If the mapped locations of the ends of a clone are “invalid” (i.e. have abnormal distance or orientation) then a genomic rearrangement is suggested. [4]

Ultra-Low RNA Input

Ultra-Low RNA Input workflow enables RNA sequencing from extremely low amounts of total RNA, starting with as little as 100 pg total RNA. It offers the powerful attributes of RNA-Seq with unparalleled sensitivity, accurate gene quantification, and dynamic range. [3]

1.2.2 RNA-Seq Advantages

Data Benefits

- Provides discreet and digital sequencing reads count that can be aligned to particular sequence.
- Capture all changes in gene expression.
- Real-time discovery.
- Applicable for any specie or transcriptome, even with no prior knowledge about the sequence. [3]

Cost

- RNA-Seq experiments are scalable for many applications.
- Get arrays with minimal reads.
- Faster results and lower cost per sample, because of high throughput of RNA-Seq.
- Constant sequencer improvements make the price per sample lower [3]

Workflow

- With an RNA-Seq library preparation workflow, the potential error can be minimized.
- Integrated indexed adapters enhance the performance.
- Eliminating gel purification reduces the time consuming.
- Possibility to automate the RNA-Seq workflow for the high volume. [3]

Software options

- Developed an open-source software.
- Possibility to reanalyze the data as a new information is available.
- Gives a digital profile of the whole transcriptome. [3]

1.3 Next-Generation Sequencing

The Next-Generation Sequencing (NGS) is a revolutionary technology provides cheap, correct and accurate information about DNA sequence. Capillary electrophoresis-based Sanger sequencing has always been constrained by few limitations in throughput; the scalability, the speed and the resolution that often prevent the scientists from obtaining the most important information, which they need for their study [5]. NGS has completely

dominated the area of basic and applied research that dealing with DNA analysis. The latest next-generation sequencing instruments can generate as much data in 24 hours as several hundred of Sanger-type DNA capillary sequencers, but they are operated by a single person [6]. Nowadays, NGS is used also in clinic diagnosis, mainly in those applications where huge quantity of sequence information or high resolution is needed.

The principle of NGS technology is similar to Sanger method; the bases of a small fragment of DNA are identified from signals emitted, as each fragment is synthesized from a DNA template strand. NGS runs this process in millions of reactions in massively parallel way, it is not limited to a single or a few DNA fragments. That is why NGS is also called massive parallel sequencing. This advantage allows rapid sequencing of large sections of DNA, such as whole genomes. The overview of whole-genome sequencing in Figure 3 shows the principle. There is a single genomic DNA (gDNA) in the first step (A). Then in the second step (B) the gDNA is fragmented into a library of small segments that are sequenced in parallel. While the individual sequence reads are compiled by aligning to a reference genome in the third step (C). Eventually in the last step (D) the whole genome sequence is derived from the consensus of aligned reads. NGS produces hundreds of gigabases of data in a single sequencing run [7].

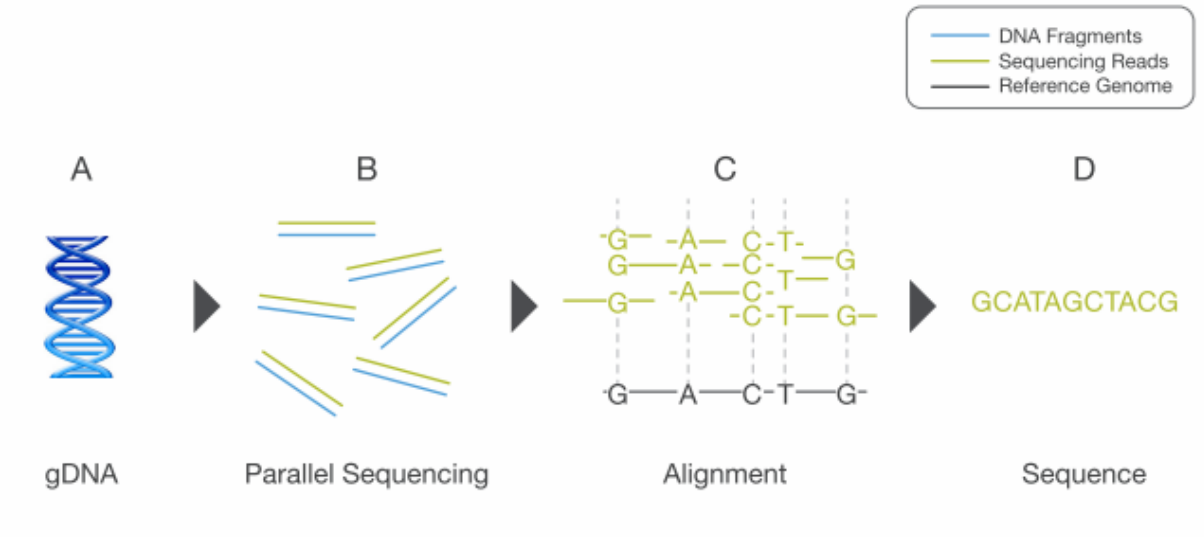


Figure 3: Overview of the whole-genome sequencing (modified from www.illumina.com)

More important than the sequencing throughput and its relative low cost compared with traditional Sanger method, is the type of data it generates. NGS provides much shorter reads (~21 to ~400 bp), but millions of them instead of long reads generated from a PCR-amplified samples. Another advantage is high flexibility for the level of resolution for a given

experiment. It can focus on specific regions of the genome with high resolution (cancer research), or provide more expansive view to the whole genome with lower resolution. To adjust the level of resolution, the coverage is tuned. The coverage means, the average number of sequencing reads that align to each base within the sample DNA. (Example: a whole genome sequenced at 25x coverage means that, on average, each base in the genome was covered by 25 sequencing reads) [6].

NGS provides quantitative data – discrete and digital sequencing read counts – it allows quantifying applications, such as gene expression analysis [16].

However, the huge amount of data from next-generation sequencing studies might take a relatively long time to be translated into useful clinical information. [9]

1.3.1 454 (Roche), 2005

It is the first commercial platform of the NGS. It is unique combination of Sanger read lengths and NGS high throughput. The system 454 uses beads; each one bead equals one DNA fragment or one read. After emulsion amplification millions of beads are loaded onto PicoTitre Plate, where the design allows to bind just one bead per one well. All beads are then sequenced in parallel using pyrosequencing reaction. [8]

The complete sequencing workflow:

1. Generating of a single-stranded DNA library.
2. Amplifying of the library using emulsion PCR (millions of copies).
3. Sequencing – data generating.
4. Data analysis.

Figure 4: emulsion PCR (www.454.com)

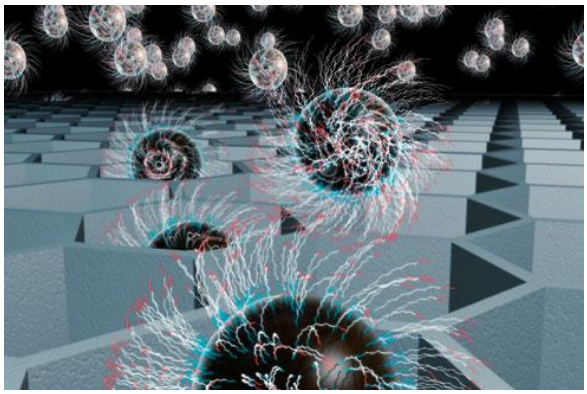


Figure 5: PicoTitrePlate (www.454.com)

Application:

- Whole genome sequencing (de novo, sequencing).
- Amplicon sequencing.
- Transcriptome sequencing (RNA-seq – gene expression analysis).
- Metagenomics.

1.3.2 Illumina (Illumina), 2007

It is the second commercial platform of the NGS. The principle is based on amplification using "bridge" PCR on solid glass surface to amplify DNA into small clusters and sequence it using synthesis. Illumina genome analyzer provides variable lengths of reads (36-300 bp), (Table 1). Illumina has the lowest cost per read per Mb. In the beginning it was for genome sequencing, but nowadays it is widely used in many other applications. [9]

Complete workflow can be divided into 3 steps:

1. Library Preparation.
2. Cluster Generation.
3. Sequencing.

Library Preparation

DNA is randomly fragmented. The ends of fragments are repaired; by adding adenine overhang (Figure 6, B) and ligating the adapters to both ends of the fragments (Firuge 6, C). Then the ligated DNA fragments are selected using gel electrophoresis (Figure 6, D).

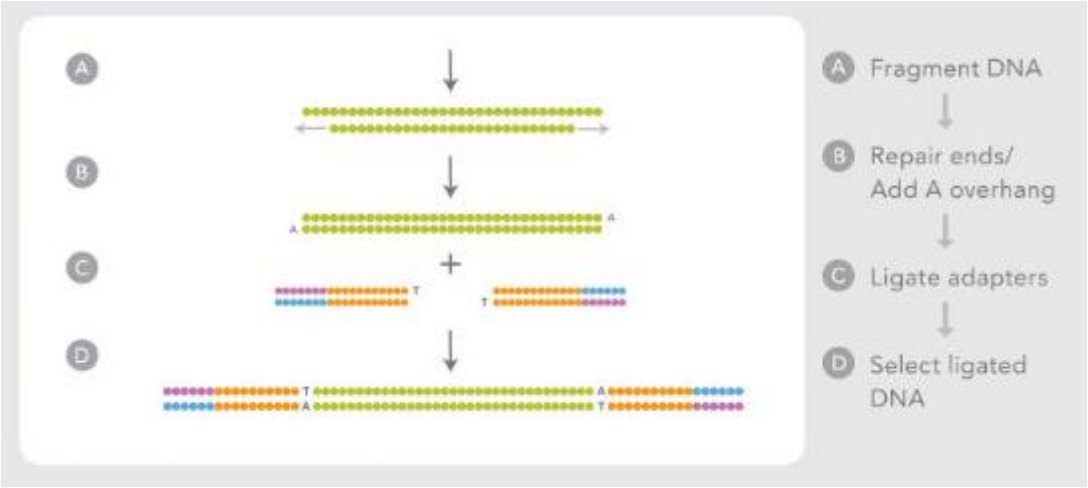


Figure 6: Illumina workflow – Library Preparation [9]

Cluster Generation

The single-strand fragments are randomly bounded to the surface (Figure 7, E). Unlabeled nucleotides and enzyme are added to perform bridge PCR (Figure 7, F). During the next step DNA fragments are amplified (Figure 7, G) and the clusters are generated (Figure 7, G). In the end the sequencing primer is annealed (Figure 7, H).

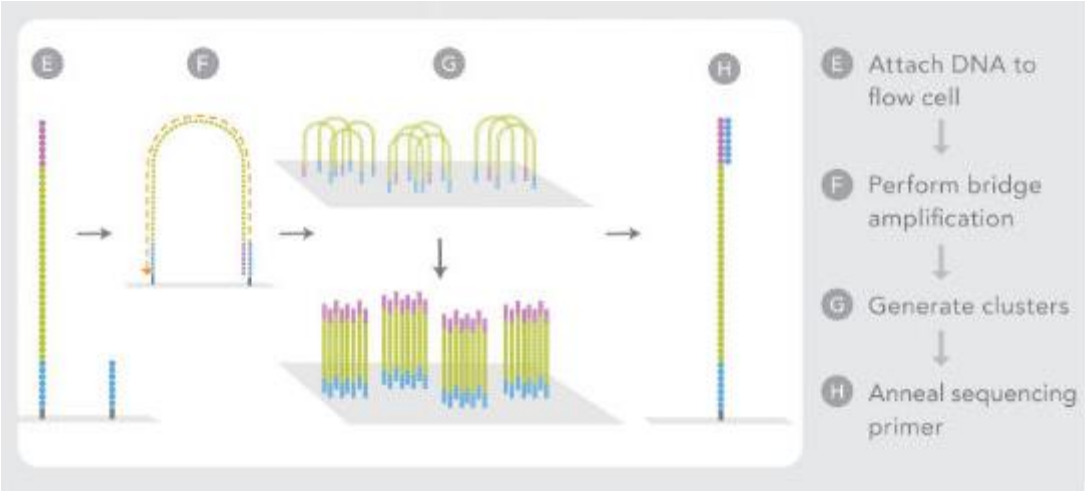


Figure 7: Illumina workflow – Cluster Generation [9]

Sequencing

To initiate the first sequencing cycle, all the four labeled reversible terminators, primers and DNA polymerase are added to the flow cell. After the laser excitation, the image of emitted fluorescence from each cluster on the flow cell is captured. The identity of the first base for each cluster is recorded (Figure 8, I). The previous step is repeated until the whole strand is extended (Figure 8, J). After generating base calls (Figure 8, K), data can be aligned, compared to a reference genome to identify the differences.

Application:

- Whole-genome sequencing.
- De novo sequencing.
- Targeted sequencing.
- DNA sequencing.
- RNA sequencing .
- Methylation analysis.

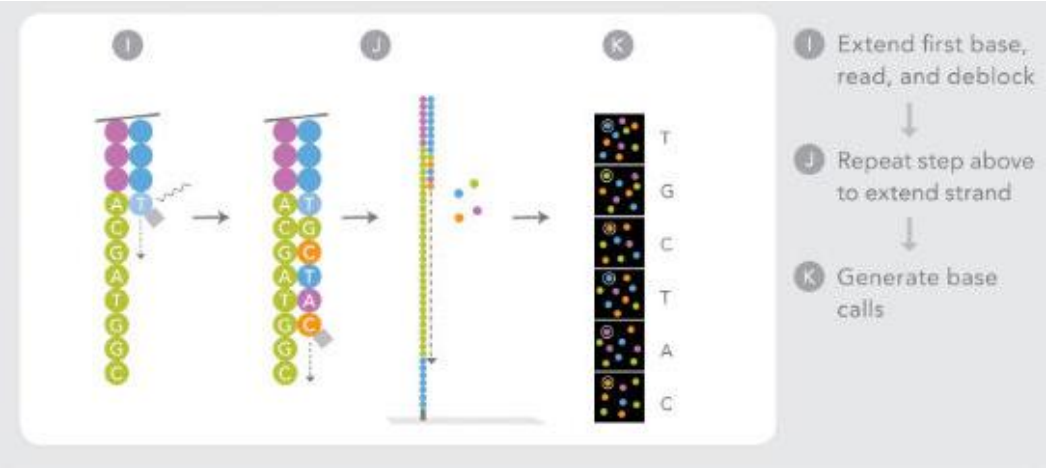


Figure 8: Sequencing steps [9]

1.3.3 SOLiD System (Applied Biosystems / Life Technologies), 2008

The massively parallel sequencing by hybridization-ligation, and the implementation in the supported oligonucleotide ligation and detection system (SOLiD), became available in 2008. The ligation chemistry is used in SOLiD, based on the same technique as in 454 [10]. Construction of libraries for analysis begins with emulsion PCR single-molecule amplification, similar to the using technique in 454. The products of amplification are transferred onto a glass surface, where the sequencing happens by sequential rounds of hybridization and ligation. The glass surface with the products is labeled by four different fluorescent colors. As the 4 colors encoding scheme is used, each position is probed twice. The identity of nucleotide is determined by analyzing the color that results from two sequential ligation reactions. This method has significantly higher specificity and a higher accuracy than the sequencing by synthesis approach [11]. SOLiD system produces 1-3 Gb of sequence data in 35-bp reads. (Table 1)

Application:

- Whole genome sequencing.
- Microbial and eukaryotic sequencing.
- Medical sequencing.
- Gene expression.
- Small RNA discovery.

1.3.4 Ion Torrent (Life Technologies), 2010

Ion semiconductor sequencing is considered to be the fastest and most affordable benchtop sequencer. The Ion Personal Genome Machine (PGM) Sequencers delivers the fastest run time, at the most affordable price, of any next-generation sequencer. High accuracy and long reads of the Ion PGM Sequencer makes the next-generation sequencing more accessible to scientists. This technology directly translates chemically encoded information (A, T, C, G) into digital information (0, 1) on a semiconductor chip. The principle of this technology that the hydrogen ion is released as a byproduct, when a nucleotide is incorporated into a DNA strand by a polymerase. It is based on the direct detection, without scanning, nor cameras, nor light. This type of detection using ion sensor makes Ion PGM Sequencer so fast. However, the throughput is currently lower than the other NGS systems. Developers hope to change it by increasing density of the chip. [12]

Complete workflow is affordable, almost fully automated:

1. Library construction.
2. Template preparation.
3. Sequencing (only hours, not days).
4. Data analysis.

As the Ion PGM Sequencer is most flexible and scalable technology, the application is wide from the targeted sequencing, through the exome sequencing, and the transcriptome sequencing to the whole genome sequencing. [12]

1.3.5 Oxford Nanopore (Oxford NANOPORE Technologies), 2012

The platform technology analyzes single molecules, it is also called the 3rd generation sequencing, as no amplification is required. Oxford Nanopore's system uses nanopore sequencing to rapidly read DNA sequences. A DNA strand is fed through a biological pore by an enzyme and the various bases are identified by measuring the difference in their electrical conductivity as they pass through the pore [13] (Figure 9). The most important advantages of this technology are the potential for dramatically longer read lengths (from tens of bases to tens of thousands of bases per read (Table 1), shorter time (from days to hours or minutes), small amounts of starting material (theoretically only single molecule) and lower overall cost. [14]

The initial system provides the nodes containing 2.000 nanopores that can read DNA at a rate of hundreds of kilobases per second. Gordon Sanghera, Oxford's chief executive said, that combining 20 nodes containing 8.000 nanopores would theoretically be able to sequence a whole human genome in 15 minutes.

The company has developed two systems. The portable MiniION device would theoretically allow doctors to sequence directly from a patient's blood in the clinic, while the larger GridION device can sequence a whole genome in a day.

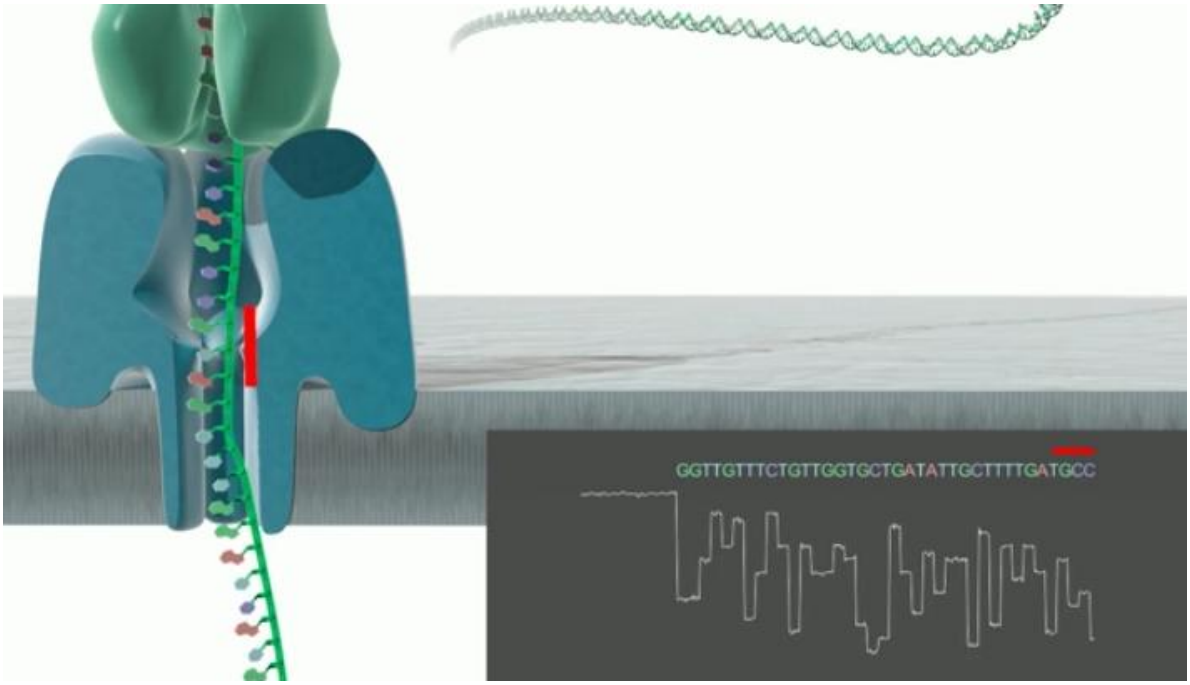


Figure 9: The nanopore sequencing identification of the bases in a DNA strand as it passes through a pore (www.nanoporetech.com)

DNA sequencing applications:

- Whole genome sequencing (de novo, sequencing).
- Targeted sequencing.
- Gene expression analysis.
- Metagenomics.

New generation sequencing methods have undergone very fast evolution in previous decade. In Table 1 are summarized the methods and their properties that discussed in this work.

Platform	*Year*	*Seq. method*	*Amplification*	*Read length*	*Detection*	*Features*
454	*2005*	*Pyro-sequencing*	*Emulsion PCR*	*200-300 bp*	*Light*	*1st NGS*
Illumina	*2007*	*Synthesis*	*Bridge PCR*	*36-300 bp*	*Light*	*90% of market*
SOLiD	*2008*	*Ligation*	*Emulsion PCR*	*35 bp*	*Light*	*Lowest Error Rate*
Ion Torrent	*2010*	*Synthesis*	*Emulsion PCR*	*400 bp*	*Hydrogen Ion*	*Semiconductor Chip*
Oxford Nanopore	*2012*	*Nanopore*	*None=Single molecule*	*>10.000 bp*	*Electrical Conductivity*	*"Run Until'' Sequencing*

Table 1: Summary of the recent NGS methods (modified from [15])

1.4 NGS Data Analysis Workflow

Next-generation sequencing has surely many advantages, especially decreasing price per Mb or whole genome, speed, high throughput and accuracy. However, the data is less understood and the data analysis is still under development. [16]

Few important questions must be answered before every experiment:

Where on the genome did each fragment originate?

For each gene, how many fragments did originate from this gene?

What are the problems we may encounter?

As the NGS brings the biggest benefit in Human diseases researches, there is a good reference genome available, and it is possible to align obtained NGS data to it [17]. Below is one of the workflows how to handle with NGS Data. (Figure 10)

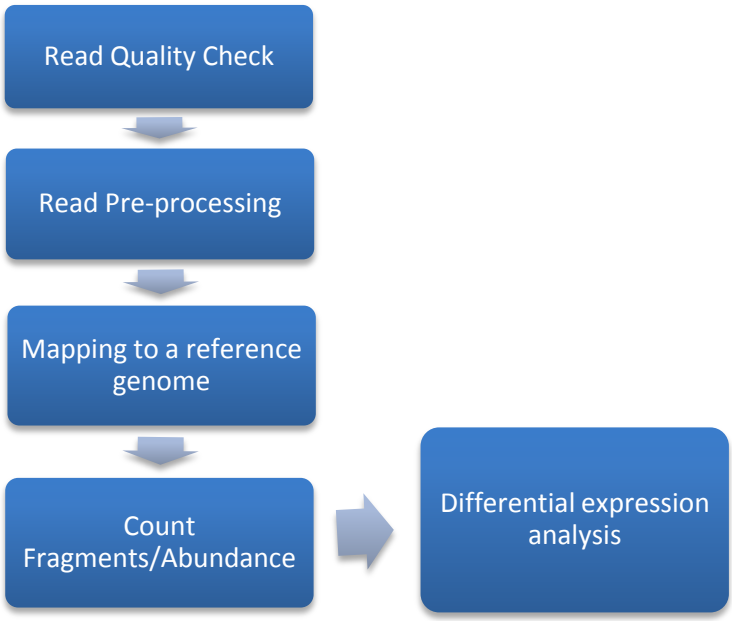


Figure 10: Data Analysis Workflow with a good reference genome [16]

1.4.1 FASTQ format

Raw NGS data is usually in FASTQ format. It has emerged as a common file format for sharing sequencing reads data and combining both the sequence and the associated quality score per base [18]. FASTQ format is practically extension of previous FASTA format, but the last-mentioned still widely used. However, the FASTA format is not ideal for very long sequences.

In FASTQ format each read has 4 lines:

1. ID
2. Reads sequence
3. Optional ID
4. Quality score

ID

ID begins with @ and includes information about the sequence.

Example:

@HWI-ST972:1044:D0E8NACXX:8:1101:1098:2055 1:N:0: ATCACG

The red is machine, run and lane identifier.

The blue: is the read (fragment) identifier.

The green: is the direction of the read 1 or 2 (3′ or 5′ end of the paired DNA reads).

The black: is the passed or the failed filter. Barcode sequence.

Read Sequence

The sequence itself

Example:

GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTC

Optional ID

It always begins with "+" and contains additional information about the sequence.

Quality score

Quality scores. The most used is the Phred Quality Scores.

1.4.2 Phred Quality Scores

Phred Quality Score of a nucleotide base is considered to be standard for estimating the probability of error. The quality value is assigned for each base. History of Phred Quality Scores goes back to Human Genome Project, where it helped in the automation of DNA sequencing. Nowadays, it is widely accepted quality format to characterize the quality of DNA sequences. Phred Quality Score can be used to compare the efficacy of different sequencing methods.

Phred Quality Score $\mathbf{Q_{PHRED}}$ is defined as a property which is logarithmically linked to the base error probabilities **P** [19].

$$Q_{PHRED} = -10 \times \log_{10}(P) \quad \textbf{(1)}$$

The quality scores are shown in Table 2.

Phred Score	Probability of incorrect base call	Accuracy
10	1 in 10	90 %
20	1 in 100	99 %
30	1 in 1000	99.9 %
40	1 in 10000	99.99 %
50	1 in 100000	99.999 %

Table 2: Phred Quality Scores and their relation to the accuracy

Currently raw Illumina data quality scores are expected in the range 0-40 [18].

Phred scores are stored as ASCII printable characters to make the file more human readable and easily edited.

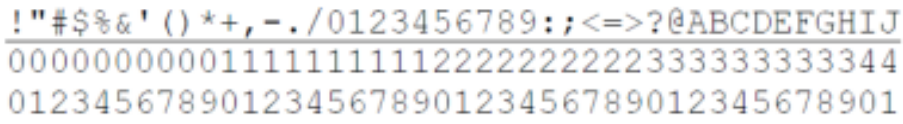


Figure 11: ASCII characters corresponding to Phred quality scores

1.5 Application and perspective of the NGS

The application of the NGS has already been described above together with every single technology of the NGS. The 2nd and 3rd generation sequencing technologies lead to more comprehensive understanding of the living systems and the phenotypes (like human disease), that emerge from this system. The 2nd generation sequencing technologies have already a huge impact on DNA sequencing. They are used to identify many rear variations in tumor tissues associated with different cancer types, as for example; the pancreatic cancer, the glioblastoma or the colon cancer [20, 21, 22]. The huge perspective of the NGS is in the personalized medicine, where the scientists are developing and using diagnostic tests based on the genetics or the other molecular mechanisms to better predict patients' responses to targeted therapy [23]. Finally, the NGS is very useful tool in the analysis of differential gene expression.

1.6 Negative Binomial model

The Negative Binomial Distribution is the distribution of the number of trials needed to get the r^{th} success and to get the fixed number of successes. We suppose there are independent trials and each trial results in one or two possible outcomes, which are labeled success and failure.

Notation: NB(r,)

Negative Binomial Distribution has two parameters:

- : Is the success probability in each experiment, 0 < < 1
- r: Is the number of failures until the experiment is stopped, $r > 0$

The probability mass function of the Negative Binomial distribution is:

(1)

We are usually interested in two characteristics of the Negative Binomial distribution:

- Mean defined as ——
- Variance defined as ——

The Figure 12 illustrates the graph of the Negative Binomial distribution for the growth of the variable *r*.

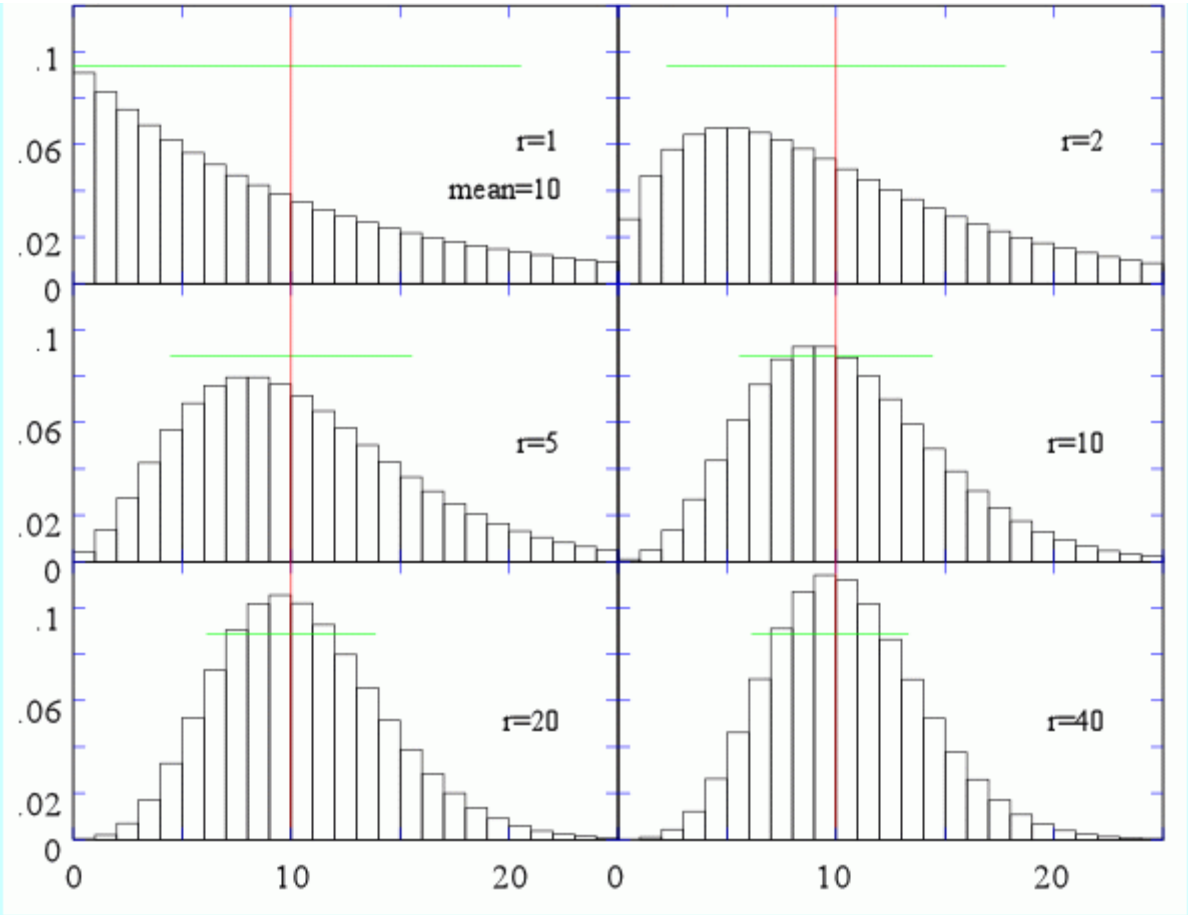


Figure 12: Negative Binomial distribution. The graphs show that for increasing r, negative binomial distribution is approaching normal distribution (http://www.eistat.cz/teorie/rozdeleni/diskretni/negbinomic/index.htm)

1.7 Prostate Cancer

Prostate cancer is a form of cancer developing in the prostate; a gland in the male productive organ. Prostate cancer is one of the most frequently diagnosed malignant tumor affecting men around the whole world. Most of the tumors are growing slowly, but there are also cases of aggressive prostate cancer [35]. The rates of detection of prostate cancers vary across the world, in South and East Asia is the rate lower than in Europe and in the United States [36]. The therapeutic success rate for the prostate cancer is higher if the disease is diagnosed in early stage. The successful therapy for this disease depends on the clinical biomarkers for early detections of the presence and progression of the disease, as well as the prediction after the clinical intervention. [37]

Signs a symptoms

The prostate cancer in initially stage causes often no symptoms, but in later stages causes pain, frequent urination, problems during sexual intercourse, erectile dysfunction and death [38]. Advanced prostate cancer can metastasize to other parts of the body and metastasis can cause new additional symptoms. The most common symptom is pain in bones (vertebrae, pelvis or ribs). The prostate cancer can also compress the spinal cord and cause leg weakness and urinary and fecal incontinence [39].

Risk factors

A complete epidemiology of prostate cancer is still not clear [40]. The most important risk factors are obesity, age and genetic. Men younger than 45 years, usually don´t suffer from this disease, the prostate cancer is more common with advancing age. The average age of men diagnosed with prostate cancer is 70 [41]. Men who have family members with prostate cancer appear to have a double risk to get this disease compared to men without prostate cancer in family. Men suffering from higher blood pressure have also higher risk [42].

Diagnosis

Prostate cancer can be diagnosed using less invasive methods or invasive biopsy, which can fully confirm the diagnosis of prostate cancer. The less invasive method is measuring the Prostate Specific Antigen (PSA) in blood samples. It is recommended to undergo a rectal examination to detect the prostate abnormalities to all men older than 45 years. Another non-invasive method is the prostate imaging. Ultrasound and Magnetic Resonance Imaging (MRI) are the most common methods used for prostate cancer detection. Ultrasound is less used for

its poor tissue resolution, the magnetic resonance imaging has better resolution [43]. For the evaluation of prostate cancer is important to determine the stage. Determining the stage helps to define the prognosis and to choose suitable therapy. The most common system is the four-stage system (Figure 13), which takes in account the size of the tumor, the number of the involved lymph nodes and the presence of another metastases [44].

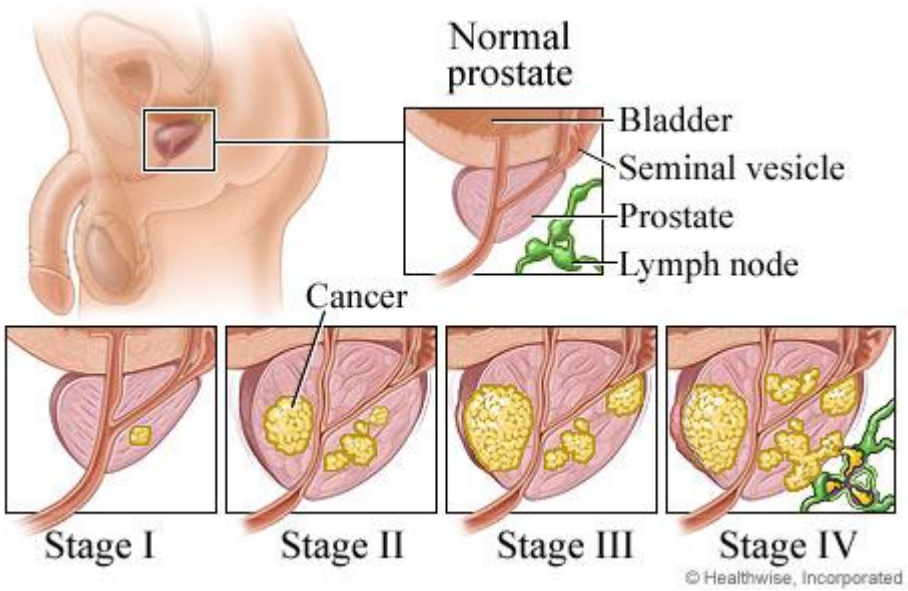


Figure 13: The four-stage system for diagnosis of prostate cancer.

Treatment

There are many options of prostate cancer treatment, the choose of the right one or the combination of few of them depends on age and expected life span of the patient, any other serious health conditions, the stage of the cancer, and the feelings of the patient about the side effects from each treatment. Treatment usually involves surgery, radiation therapy, proton therapy, and less commonly; cryotherapy, hormone therapy and chemotherapy, which are usually recommended for advanced stages of the disease. All treatments can have significant side effects (erectile dysfunction or urinary incontinence), it is important to find the balance between the goals of the therapy and the risks of lifestyle alternations. Doctors usually recommend the combination of treatment methods [45]. Androgen deprivation therapy (ADT) plays major role in the treatment of prostate cancer, but needs accurate timing. Overuse of ADT is helpful to avoid the side effects of castration, which are the effects on sexual function, bone mineral density, lipid metabolism and insulin sensitivity that influence increasing morbidity and decreasing quality of life [46]. There is no evidence of the benefit from ADT for localized cancer, but studies have reported an increased use of castration as a primary treatment for localized disease [46].

2 PRACTICAL APPLICATION

In this part, we use the open access available RNA-seq data, and analyze them by MATLAB environment (version R2013b) with the support of Bioinformatics Toolbox and the Statistics Toolbox functions. Then we test the differential gene expression using a negative binomial model.

2.1 The Prostate Cancer Data Set

The data was downloaded from Web Sites of Sanford Consortium for Regenerative Medicine ([24], http://yeolab.ucsd.edu/yeolab/Papers.html)

Li et al. published the prostate cancer study, where the prostate cancer cell line LNCap was treated with androgen/DHT. They used a double-random priming method for deep sequencing to profile double poly(A)-selected RNA from LNCaP cells before and after androgen stimulation. In this study, they uncovered 71% (from 20 million sequence tags) of annotated genes and identified hormone-regulated gene expression events that are significantly correlated with the quantitative real time PCR measurement. A fraction of the sequence tags were mapped to constitutive and alternative splicing events to detect known or new mRNA isoforms expressed in the prostate cancer cell. In the end, they used curve fitting to estimate the number of tags necessary to reach a saturating discovery rate among individual applications [24].

Mock-treated and androgen-stimulated LNCap cells were sequenced using the Illumina 1G Genome Analyzer. Analysis of ~10 million sequence tags generated from both mock-treated and hormone-treated cells indicates that this tag density was sufficient for quantitative analysis of gene expression [24].

For the mock-treated cells, there were four lanes totaling ~10 million reads. For the DHT-treated cells, there were three lanes totaling ~7 million reads. All replicates were technical replicates. Samples labeled s1 through s4 are from mock-treated cells. Samples labeled s5, s6, and s8 are from DHT-treated cells. The reads sequence are stored in FASTA format.

2.2 Mapping the reads

SAM/formatted files for each of the seven FASTA files were produced by mapping the reads to the human genome, version hg19, GRCh37 using a *Bowtie* aligner on Linux. The human genome was downloaded from the UCSC Genome Browser website.

2.2.1 The UCSC Genome Browser

As the vertebrate genome sequences become complete and research refocuses to their analysis, the issue of effective genome annotation display becomes problematic. There is mature web tool for rapid and reliable display of any requested portion of the genome at any scale, together with several dozen of aligned annotation tracks at http://genome.ucsc.edu. This provides displaying assembly contigs and gaps, mRNA and expressed sequence tags alignment, multiple genes prediction, cross-species homologies, single nucleotide polymorphisms, sequence-tagged sites, radiation hybrid data, transposon repeats, and more as a stack of co-registered tracks. Text and sequence-based investigations provide quick and precise access to any region of specific interest. Secondary links from individual features lead to sequence details and supplementary of other off-site databases. One-half of the annotation tracks are computed at the University of California, Santa Cruz from publicly available sequence data; collaborators worldwide provide the rest. Users can stably add their own custom tracks to the browser for educational or research purposes. [30]

To use a browser, follow the "browser" link at http://genome.ucsc.edu. This will take you to a page where you can search for a gene by name, author, keyword, and so forth. Or directly specify the region to view as either a chromosome band or a chromosome and range of bases. It is also possible to enter the browser via a search for homologous regions to a DNA or protein sequence using the "BLAT" link. The BLAT search takes typically only a few seconds, it is big advantage. The main browser contains three main parts (Figure 14). On the top is a series of controls for searching and for zooming and scrolling across a chromosome. In the middle is a dynamically generated picture that graphically displays genome annotations. On the bottom is another series of controls that fine-tune the graphic display [30]. The UCSC Genome Browser provides many options; you can use the drop-down control to alter the displayed tracks, for example: Mapping and Sequencing, Genes and Gene Predictions, mRNA and EST, Expression, Regulation and many others. Tracks of interest will be displayed automatically in more compact modes.

The UCSC Genome Browser is very useful tool for exploring the human genome. There is a possibility to download the whole human genome and work with it.

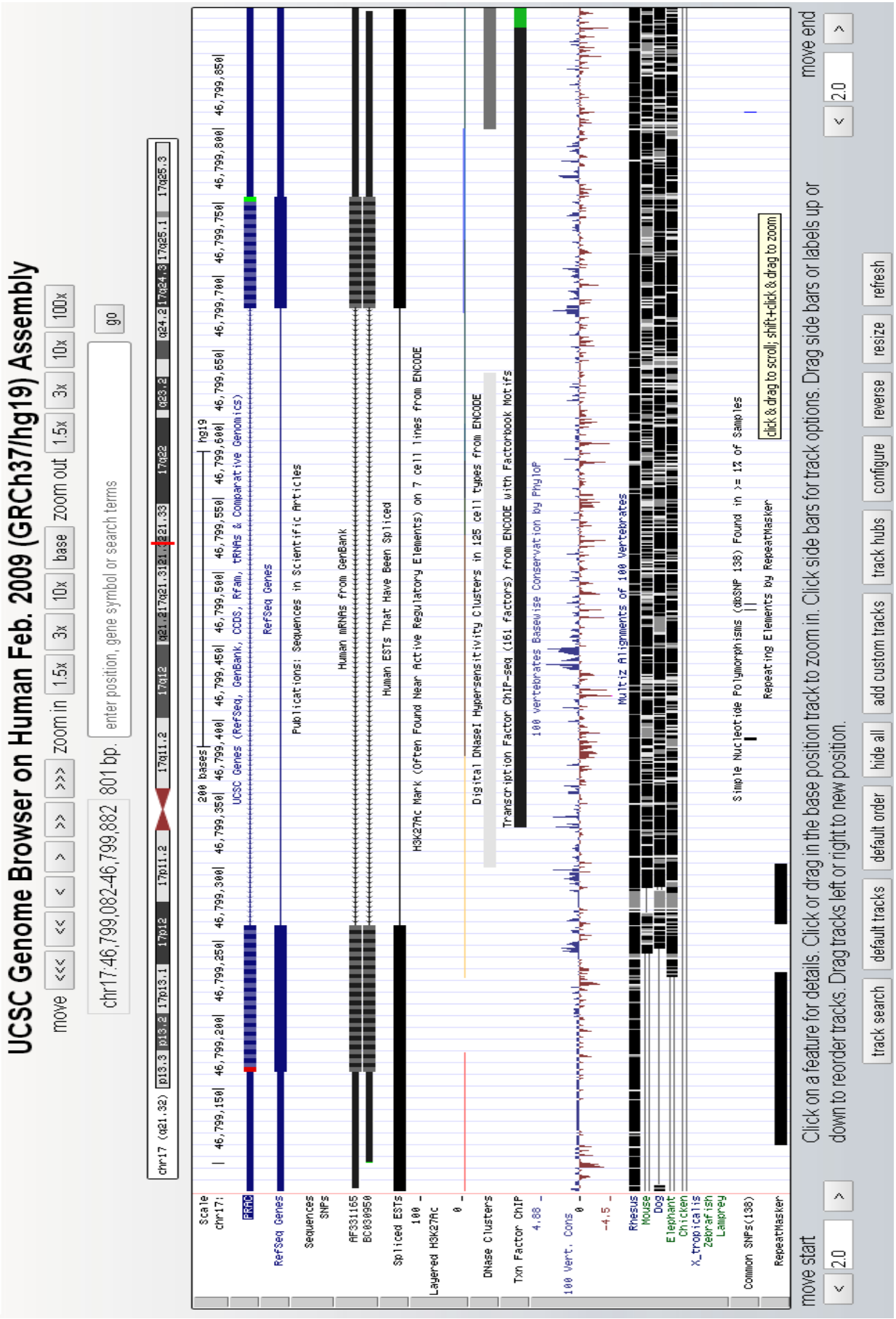


Figure 14: Part of the human chromosome 17 in the UCSC Genome Browser. This region contains gene PRAC (Homo sapiens prostate cancer susceptibility candidate). The spliced EST track indicates that there is active transcription. The Rhesus Blat track indicated a high level of conservation between Rhesus and human in this region.

2.2.2 hg19

Human genome projects have generated an unexpected amount of knowledge about human genetics and health. There is always a need to map the new knowledge to the whole human genome, which must be continuously updated. We use the human genome hg19, version GRCh37, which was produced on February 2009 by the Genome Reference Consortium [30].

Detailed information about the human genome GRCh37 is in Table 3.

Organism name	Homo sapiens
Submitter	Genome Reference Consortium
Date	27. 2. 2009
Synonyms	hg19
Assembly type	Haploid-with-alt-loci
Assembly level	Chromosome
Genome representation	full

Table 3: Detailed information about the human genome version GRCh37 (NCBI Assembly database)

Global statistics for this version are in Table 4.

Number of regions with alternate loci or patches	7
Total sequence length	3,137,144,693
Total assembly length	239,852,888
Gaps between scaffolds	271
Number of scaffolds	258
Scaffolds N50	46,395,641
Number of contigs	461
Contig N50	38,440,852
Total number of chromosomes and plasmids	24

Table 4: Global statistics of hg19 (NCBI Assembly database)

2.2.3 SAM/BAM file

The Sequence Alignment/Map-SAM format is a generic alignment text format for storing read alignments against a reference sequences, it supports both short and long reads, which are produced by different sequencing platforms [25]. With the revolution of next/generation sequencing methods (Illumina, SOLiD, 454), many of new alignment tools have been developed to realize read mapping to large reference genome, including the human genome. SAM format has become a common alignment format that supports all sequence types and creates a well-defined interface between alignment and post-processing analysis. The SAM tools provide utilities to manipulate alignments in the SAM format.

The SAM format consists of header section and alignment section. The header section starts with '@'. In this format, each alignment line has 11 mandatory fields and variable number of optional fields [25]. (Table 5)

Number	Name	Description
1	QNAME	Query name of the read
2	FLAG	Bitwise FLAG (pairing, strand…)
3	RNAME	Reference sequence name
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred Score)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (Phred Score)

Table 5: Mandatory fields in the SAM format (modified from [25])

CIGAR is standard defining pairwise alignment. It is important to know where the changes in the sequence are. It defines following operation (Table 6):

Operation	Description
M	Alignment match (can be match or mismatch)
I	Insertion to the reference
D	Deletion from the reference
N	Skipped region from the reference
S	Soft clipping (clipped sequence present in SEQ)
H	Hard clipping (clipped sequence NOT present in SEQ)
P	Padding (silent deletion from padded reference)
=	Sequence match
X	Sequence mismatch

Table 6: CIGAR operations (modified from [26])

BAM file is binary SAM. BAM uses compression scheme (BGZF-Blocked GNU Zip Format) to make alignments more compact. BAM files can be sorted and indexed, it makes accessing data very fast. SAMtools enables to view a content of BAM formatted file. BGZF is block compression implemented on top of the standard gzip file format [25]. The aim of using BGZF is to provide good compression while allowing efficient random access to the BAM file for indexed issues. [25]

2.2.4 SAMtools

SAMtools provide various opportunities for manipulating with short DNA sequence read alignments in the SAM or BAM format including sorting, merging, indexing. [25]
SAMtools provides the following commands: view, sort, index, tview and mpileup.

view: the view command converts the SAM format to BAM format

sort: the sort command sorts a BAM file based on position in the reference, as determined by its alignment.

index: the index command creates a new index file that allows fast look-up of data in a sorted a SAM or BAM file.

tview: the tview command starts an interactive viewer based on ascii that can be used to visualize how reads are aligned to specified regions of the reference genome.

mpileup: the mpileup command produces pileup format file.

SAMtools is open source, it is available at http://sourceforge.net/projects/samtools/files/.

2.2.5 BowtieBuild

Bowtiebuild is a function implemented in Bioinformatics Toolbox in Matlab. It generates index using Burrows-Wheeler transform.

Burrows-Wheeler transform

Burrows-Wheeler transform (BWT) is a method used for the lossless data compression. In general, it is a reversible permutation of the characters in a text [27]. BWT is also called as block-sorting data compression. Previously, BWT was used for text compression algorithms. However, as the high-throughput sequencing methods exploded, the BWT was used in programs for alignment NGS reads to whole genomes. The aim of using BWT in *bowtie* is reducing the memory consuming.

Syntax

`bowtiebuild(input,indexBaseName)` builds an index using the reference sequence(s) in `input` an saves it to the index file `indexBaseName`.

Usage:

To build an index of human genome hg19 we were using this command:

```
bowtiebuild('hg19.fas', 'hg19')
```

The indexes hg19.1.ebwt, hg19.2.ebwt, hg19.3.ebwt and hg19.4.ebwt were produced.

2.2.6 Bowtie

Bowtie is an ultrafast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small: typically about 2.2 GB for the human genome (2.9 GB for paired-end). [27] Bowtie can output alignments in SAM format, it allows us interoperation with other tools supporting SAM formatted files, for example SAMtools.

Bowtie runs on command line under Windows, Mac OS X, Linux and Solaris.

Bowtie is open source. It is available at http://bowtie.cbcb.umd.edu. [27]

Bowtie usage:

bowtie [options]* <ebwt> {-1 <m1> -2 <m2> | --12 <r> | <s>} [<hit>]

Bowtie options:

<m1> Comma-separated list of files containing upstream mates (or the sequences themselves, if -c is set) paired with mates in <m2>

<m2> Comma-separated list of files containing downstream mates (or the sequences themselves if -c is set) paired with mates in <m1>

<r> Comma-separated list of files containing Crossbow-style reads. Can be a mixture of paired and unpaired. Specify "-" for stdin.

<s> Comma-separated list of files containing unpaired reads, or the sequences themselves, if -c is set. Specify "-" for stdin.

<hit> File to write hits to (default: stdout)

Input:

-q	query input files are FASTQ .fq/.fastq (default)
-f	query input files are (multi-)FASTA .fa/.mfa
-r	query input files are raw one-sequence-per-line
-c	query sequences given on cmd line (as <mates>, <singles>)
-C	reads and index are in colorspace
-Q/--quals <file>	QV file(s) corresponding to CSFASTA inputs; use with -f -C
--Q1/--Q2 <file>	same as -Q, but for mate files 1 and 2 respectively
-s/--skip <int>	skip the first <int> reads/pairs in the input
-u/--qupto <int>	stop after first <int> reads/pairs (excl. skipped reads)
-5/--trim5 <int>	trim <int> bases from 5' (left) end of reads
-3/--trim3 <int>	trim <int> bases from 3' (right) end of reads
--phred33-quals	input quals are Phred+33 (default)
--phred64-quals	input quals are Phred+64 (same as --solexa1.3-quals)
--solexa-quals	input quals are from GA Pipeline ver. < 1.3
--solexa1.3-quals	input quals are from GA Pipeline ver. >= 1.3
--integer-quals	qualities are given as space-separated integers (not ASCII)

Alignment:

-v <int>	report end-to-end hits w/ <=v mismatches; ignore qualities
or	
-n/--seedmms <int>	max mismatches in seed (can be 0-3, default: -n 2)
-e/--maqerr <int>	max sum of mismatch quals across alignment for -n (def: 70)
-l/--seedlen <int>	seed length for -n (default: 28)

--nomaqround	disable Maq-like quality rounding for -n (nearest 10 <= 30)
-I/--minins <int>	minimum insert size for paired-end alignment (default: 0)
-X/--maxins <int>	maximum insert size for paired-end alignment (default: 250)
--fr/--rf/--ff	-1, -2 mates align fw/rev, rev/fw, fw/fw (default: --fr)
--nofw/--norc	do not align to forward/reverse-complement reference strand
--maxbts <int>	max # backtracks for -n 2/3 (default: 125, 800 for --best)
--pairtries <int>	max # attempts to find mate for anchor hit (default: 100)
-y/--tryhard	try hard to find valid alignments, at the expense of speed
--chunkmbs <int>	max megabytes of RAM for best-first search frames (def: 64)

Reporting:

-k <int>	report up to <int> good alignments per read (default: 1)
-a/--all	report all alignments per read (much slower than low -k)
-m <int>	suppress all alignments if > <int> exist (def: no limit)
-M <int>	like -m, but reports 1 random hit (MAPQ=0); requires --best
--best	hits guaranteed best stratum; ties broken by quality
--strata	hits in sub-optimal strata aren't reported (requires --best)

Output:

-t/--time	print wall-clock time taken by search phases
-B/--offbase <int>	leftmost ref offset = <int> in bowtie output (default: 0)
--quiet	print nothing but the alignments
--refout	write alignments to files refXXXXX.map, 1 map per reference
--refidx	refer to ref. seqs by 0-based index rather than name
--al <fname>	write aligned reads/pairs to file(s) <fname>
--un <fname>	write unaligned reads/pairs to file(s) <fname>
--max <fname>	write reads/pairs over -m limit to file(s) <fname>
--suppress <cols>	suppresses given columns (comma-delim'ed) in default output
--fullref	write entire ref name (default: only up to 1st space)

Colorspace:

--snpphred <int>	Phred penalty for SNP when decoding colorspace (def: 30)
or	
--snpfrac <dec>	approx. fraction of SNP bases (e.g. 0.001); sets --snpphred
--col-cseq	print aligned colorspace seqs as colors, not decoded bases
--col-cqual	print original colorspace quals, not decoded quals
--col-keepends	keep nucleotides at extreme ends of decoded alignment

SAM:

-S/--sam write hits in SAM format

--mapq <int> default mapping quality (MAPQ) to print for SAM alignments

--sam-nohead supppress header lines (starting with @) for SAM output

--sam-nosq supppress @SQ header lines for SAM output

--sam-RG <text> add <text> (usually "lab=value") to @RG line of SAM header

Performance:

-o/--offrate <int> override offrate of index; must be >= index's offrate

-p/--threads <int> number of alignment threads to launch (default: 1)

--mm use memory-mapped I/O for index; many 'bowtie's can share

--shmem use shared mem for index; many 'bowtie's can share

Other:

--seed <int> seed for random number generator

--verbose verbose output (for debugging)

--version print version information and quit

-h/--help print this usage message

2.2.7 Map and sort the reads

We were using Bowtie version 1.0.0 on Linux. To make mapping and sorting the reads faster and efficient, following script containing more steps in one was written and let run on command line:

```
file=s1

bowtie -f -p 2 -v 2 -m 1 --best -S hg19 $file.fa > $file.sam
samtools view -Sb $file.sam > $file.bam
samtools sort $file.bam $file.sort
samtools view $file.sort.bam > $file.sort.sam
```

The variable `file` was changed every cycle from s1 to s8.

The bowtie aligner was instructed to:

- Input files were FASTA, .fa (-f)
- Report one best valid alignment. (--best)
- No more than two mismatches were allowed for alignment. (-v 2)
- Reads with more than one reportable alignment were suppressed, i.e. any read that mapped to multiple locations was discarded. (-m 1)

First, the alignment output was seven SAM files (s1.sam, s2.sam, s3.sam, s4.sam, s5.sam, s6.sam, s8.sam). We converted these seven files to BAM files using samtools. Then sorted these BAM files and the output was seven sorted BAM files (s1.sort.bam, s2.sort.bam, s3.sort.bam, s4.sort.bam, s5.sort.bam, s6.sort.bam, s8.sort.bam). In the end we converted the sorted BAM files to SAM files and the output was again seven SAM files (s1.sort.sam, s2.sort.sam, s3.sort.sam, s4.sort.sam, s5.sort.sam, s6.sort.sam, s8.sort.sam).

2.3 Creating an annotation object of target genes

Using Ensembl´s BioMart service (http://www.ensembl.org/biomart), we can download a tab-separated-value (TSV) table with all protein encoding genes to a text file, ensemblmart_genes_hum37.txt. We are using Ensamble release 74. The table contains following attributes: chromosome name, gene biotype, gene name, gene start/end, and strand direction.

2.3.1 Ensembl BioMart

The Ensembl project was started in 1999, few years before the human genome was completely sequenced in the frame of Human Genome Project. The aim of this project was to automatically annotate the genome, integrate this annotation with other available biological data and make it all publicly available. The website was launched in July 2000 and many more genomes have been added, the range of available data has expanded, it includes comparative genomics, variation and regulatory data [28].
BioMart Project is Bio Portal including 46 databases located in 4 continents and it is still growing. This project provides free software and data services to the international scientific community. Ensemble supports downloading many correlation tables using highly customizable BioMart data mining tool [29].

After downloading a TSV table with all protein encoding genes, we used function `ensemblmart2gff` to convert the TSV file to a GFF formatted file. The GFF (General Feature Format) consists of one line per feature, each one contains 9 columns of data. The GFF file was loaded to MATLAB using function `GFFAnnotation`.

```
GFFfilename = ensemblmart2gff('ensemblmart_genes_hum37.txt');
genes = GFFAnnotation(GFFfilename)
```

```
genes =

  GFFAnnotation with properties:

    FieldNames: {1x9 cell}
    NumEntries: 22836
```

We created a subset of the genes presented in the chromosomes only. The GFFAnnotaion object contains 20327 annotated protein-coding genes in the Ensembl database.

```
chrs =
{'1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','
16','17','18','19','20','21','22','X','Y','MT'};
genes = getSubset(genes,'reference',chrs)


genes =

  GFFAnnotation with properties:

    FieldNames: {1x9 cell}
    NumEntries: 20327
```

The gene information is now in a structure, we can display the first entry.

```
getData(genes,1)

ans =

     Reference: '15'
         Start: 20737094
          Stop: 20747114
       Feature: 'GOLGA6L6'
        Source: 'protein_coding'
         Score: '0.0'
        Strand: '-'
         Frame: '.'
    Attributes: ''
```

The annotation object of targeted genes is now ready for the following tasks.

2.4 Importing Mapped Short Read Alignment Data

The size of the sorted SAM files in our data-set is in the range of 250-360 MB. We can access the mapped and sorted reads in s1.sort.sam by creating a BioMap. It is a class that has an interface, and provides direct access to the mapped and sorted short reads in SAM-formatted file, it is minimizing the amount of data that is actually loaded into the memory. BioMap includes headers, read sequences, quality scores of the sequences and information about how each sequence aligns to a given reference.

```
    bm1 = BioMap('s1.sort.sam')

bm1 =

  BioMap with properties:

    SequenceDictionary: {1x25 cell}
             Reference: [458367x1 File indexed property]
             Signature: [458367x1 File indexed property]
                 Start: [458367x1 File indexed property]
       MappingQuality: [458367x1 File indexed property]
                  Flag: [458367x1 File indexed property]
          MatePosition: [458367x1 File indexed property]
               Quality: [458367x1 File indexed property]
              Sequence: [458367x1 File indexed property]
                Header: [458367x1 File indexed property]
                 NSeqs: 458367
                  Name: ''
```

After creating the BioMap object, we used the getSummary method to obtain a list of the existing references and the actual number of the short read mapped to each one. We observed that the order of the references is equivalent to the previously created cell string `chrs`.

```
getSummary(bm1)
BioMap summary:
                                  Name: ''
                        Container_Type: 'Data is file indexed.'
              Total_Number_of_Sequences: 458367
```

Number_of_References_in_Dictionary: 25

	Number_of_Sequences	Genomic_Range	
chr1	39037	564571	249213991
chr2	23102	39107	243177977
chr3	23788	578280	197769619
chr4	16273	56044	190988830
chr5	20875	50342	180698591
chr6	16743	277774	170892222
chr7	17022	146474	158834423
chr8	12199	162668	146284742
chr9	13988	21790	141067447
chr10	15707	179281	135500747
chr11	37506	203411	134375386
chr12	21714	79745	133785475
chr13	6078	19335895	115091858
chr14	14644	19123810	107260517
chr15	13199	20145084	102501644
chr16	15423	92212	90143169
chr17	22089	56680	81014350
chr18	5986	111538	77957293
chr19	17690	63006	59093541
chr20	10026	119233	62906673
chr21	6119	9421584	48085597
chr22	7366	16150315	51216589
chrX	12939	2774622	154563685
chrY	2819	2711686	59032821
chrM	66035	12	16570

2.5 Determining Digital Gene Expression

We determined the mapped reads associated with each Ensembl gene.

The reference names in the SAM files are different to those provided in the annotations, we found a vector with the reference index for each gene:

```
geneReference =  seqmatch(genes.Reference,chrs, 'exact',true);
```

Then we counted the mapped reads that overlap any part of the gene for each one. The reads count for each gene is the digital gene expression of that gene. We used the `getCounts` method of a `BioMap` to compute the reads count within a specified range.

```
counts1 =
getCounts(bm1,genes.Start,genes.Stop,1:genes.NumEntries,geneReference);
```

Levels of gene expression can be better represented by a `DataMatrix`, where each row represents a gene, and each column represents a sample. We created a `DataMatrix` with seven columns, one for each sample. Using these commands, we copied the counts of the first sample to the first column.

```
filenames = {'s1.sort.sam', 's2.sort.sam', 's3.sort.sam',
's4.sort.sam', 's5.sort.sam', 's6.sort.sam', 's8.sort.sam'};
samples =  {'Mock_1', 'Mock_2', 'Mock_3', 'Mock_4', 'DHT_1', 'DHT_2',
'DHT_3'};

lncap_counts =
bioma.data.DataMatrix(NaN([genes.NumEntries,7]),genes.Feature,samples)
lncap_counts(:,1) = counts1
```

We displayed ten counts of genes from 190 to 200:

```
lncap_counts(190:200,:)
ans =
```

	Mock_1	Mock_2	Mock_3	Mock_4	DHT_1	DHT_2	DHT_3
FAM217B	20	NaN	NaN	NaN	NaN	NaN	NaN
PTPRA	75	NaN	NaN	NaN	NaN	NaN	NaN
PPP1R3D	1	NaN	NaN	NaN	NaN	NaN	NaN
SNRPN	269	NaN	NaN	NaN	NaN	NaN	NaN
NR6A1	8	NaN	NaN	NaN	NaN	NaN	NaN
CDH26	43	NaN	NaN	NaN	NaN	NaN	NaN
RGL4	21	NaN	NaN	NaN	NaN	NaN	NaN
CTSA	6	NaN	NaN	NaN	NaN	NaN	NaN

```
    OLFML2A      0       NaN       NaN       NaN       NaN       NaN       NaN
    USP16       32       NaN       NaN       NaN       NaN       NaN       NaN
    ESCO1       16       NaN       NaN       NaN       NaN       NaN       NaN
```

Then we determined the number of genes that have counts greater or equal to 50 in chromosome 1:

```
lichr1 = geneReference == 1;  % logical index to genes in chromosome 1
sum(lncap_counts(:,1) >= 50 & lichr1)

ans =

   189
```

In the end, we repeated this step for other six samples in the data-set to get their gene counts and copy the information to the previously created `DataMatrix`.

```
for i = 2:7
    bm = BioMap(filenames{i});
    counts =
getCounts(bm,genes.Start,genes.Stop,1:genes.NumEntries,geneReference);
    lncap_counts(:,i) = counts;
end
```

Now the `DataMatrix` is completed, we can again display the genes from 190 to 200.

```
>> lncap_counts(190:200, :)
ans =

              Mock_1    Mock_2    Mock_3    Mock_4    DHT_1     DHT_2     DHT_3
    FAM217B     20        18        34        31        24        25        21
    PTPRA       75        87       106       109       110       130        38
    PPP1R3D      1         3         1         2         3         8         4
    SNRPN      269       315       366       364       422       447       138
    NR6A1        8         8        15        18         7        13         2
    CDH26       43        34        47        53         3         3         0
    RGL4        21        20        31        28        24        22        13
    CTSA         6         5         8         7         8         5         0
    OLFML2A      0         0         0         0         0         0         0
    USP16       32        42        33        55        86        94        29
    ESCO1       16        22        35        40       100       107        39
```

The `DataMatrix` "`lncap_counts`" contains counts of samples from two different biological conditions: mock-treated and DHT-treated.

```
cond_Mock = logical([1 1 1 1 0 0 0]);
cond_DHT  = logical([0 0 0 0 1 1 1]);
```

We can easily plot the counts for a chromosome along the genome coordinate. We created a plot of the counts for chromosome 1 for mock-treated sample Mock_1 and DHT-treated sample DHT_1. We added the ideogram for chromosome 1 to the plot using the `chromosomeplot` function.

```
ichr1 = find(lichr1);  % linear index to genes in chromosome 1
[~,h] = sort(genes.Start(ichr1));
ichr1 = ichr1(h);      % linear index to genes in chromosome 1 sorted by
                       % genomic position

figure
plot(genes.Start(ichr1), lncap_counts(ichr1, 'Mock_1'), '.-r',...
     genes.Start(ichr1), lncap_counts(ichr1, 'DHT_1'), '.-b');
ylabel('Gene Counts')
title('Gene 'Gene Counts on Chromosome 1')
fixGenomicPositionLabels(gca)  % formats tick labels and adds datacursors
chromosomeplot('hs_cytoBand.txt', 1, 'AddToPlot', gca)
```

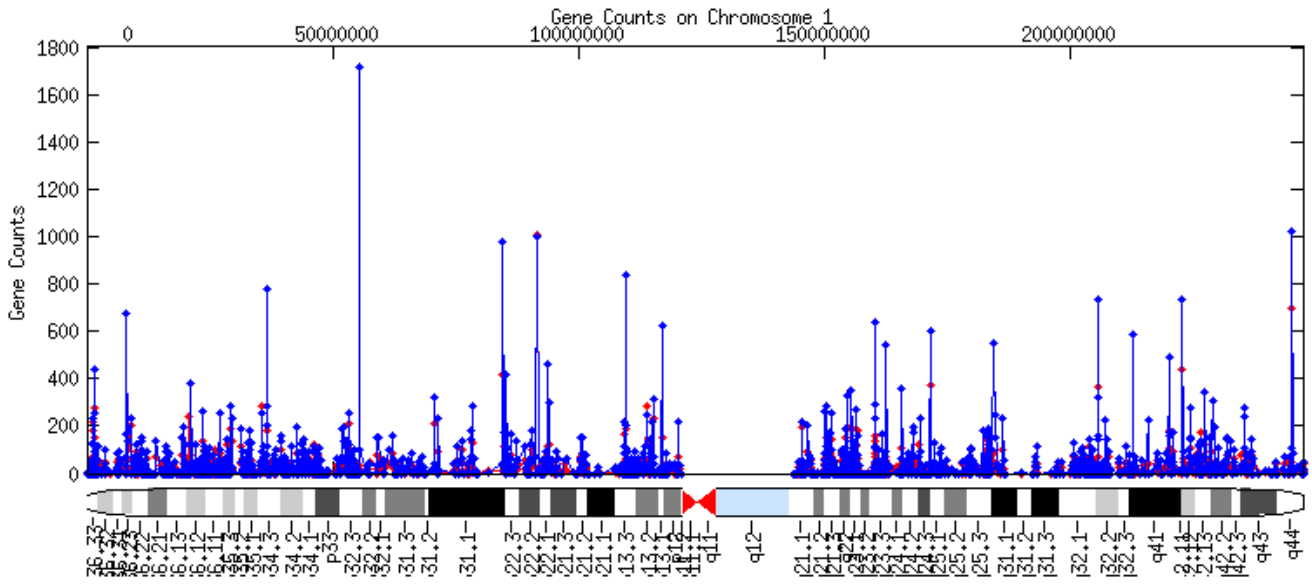


Figure 15: Plot of the counts for chromosome 1. The counts of DHT-treated sample are shown in blue, and the counts of Mock-treated sample are shown in red.

We can already see in Figure 15, that some of the genes are up-regulated and some of them are down-regulated.

2.6 Inference of Differential Signal in RNA Expression

The reads counts for RNA-seq experiments have been found to be linearly related to the abundance of the targeted transcripts [31]. It is interesting to compare the reads counts between different biological conditions. Current references suggest that typical RNA-seq experiments have quite low background noise; the gene counts are discrete and can follow the Poisson distribution. It was noted that the presumption of the Poisson distribution often predicts smaller variation in count data by ignoring the extra variation due to the actual differences between replicate samples [32]. Anders et. al., (2010) designed an error model for statistical inference of differential signal in RNA-seq expression data that could cause the overdispersion problems [33]. Their model uses the negative binomial distribution to model the null distribution of the reads counts. The variance and mean of the negative binomial distribution are linked by local regression, these two parameters can be well estimated even if the number of replicates is small [33].

In our project, we applied the Negative Binomial distribution to process the data count and test for the differential expression. The model we are following, the Anders´s model has three sets of parameters that need to be estimated from the data-set:

1. Library size parameters;
2. Gene abundance parameters under each experimental condition;
3. The smooth functions that model the dependence of the raw variance on the expected mean.

2.7 Estimating Library Size Factor

The values of all genes counts from a sample that we are expecting are proportional to the sample´s library size. The effective library size was estimated from the data count.

We computed the geometric mean of the gene counts (rows in lncap_counts) across all samples in the experiment as a pseudo-reference sample:

```
geoMeans = exp(mean(log(lncap_counts), 2));
```

Then we computed each library size parameter as the median of the ratio of the sample´s counts to those of the pseudo-reference sample:

```
ratios = dmbsxfun(@rdivide, lncap_counts(geoMeans >0, :),
geoMeans(geoMeans >0));
sizeFactors = median(ratios, 1);
```

The counts were transformed to a common scale using size factor adjustment:

```
base_counts = dmbsxfun(@rdivide, lncap_counts, sizeFactors);
```

Now we can use the `boxplot` function to inspect the count distribution of the mock-treated and DHT-treated samples and the size factor adjustment:

```
figure
subplot(2,1,1)
maboxplot(log2(lncap_counts), 'title', 'Raw Read Counts',...
                              'orientation', 'horizontal')
subplot(2,1,2)
maboxplot(log2(base_counts), 'title', 'Size Factor Adjusted Read
                                                    Counts',...
                              'orientation', 'horizontal')
```

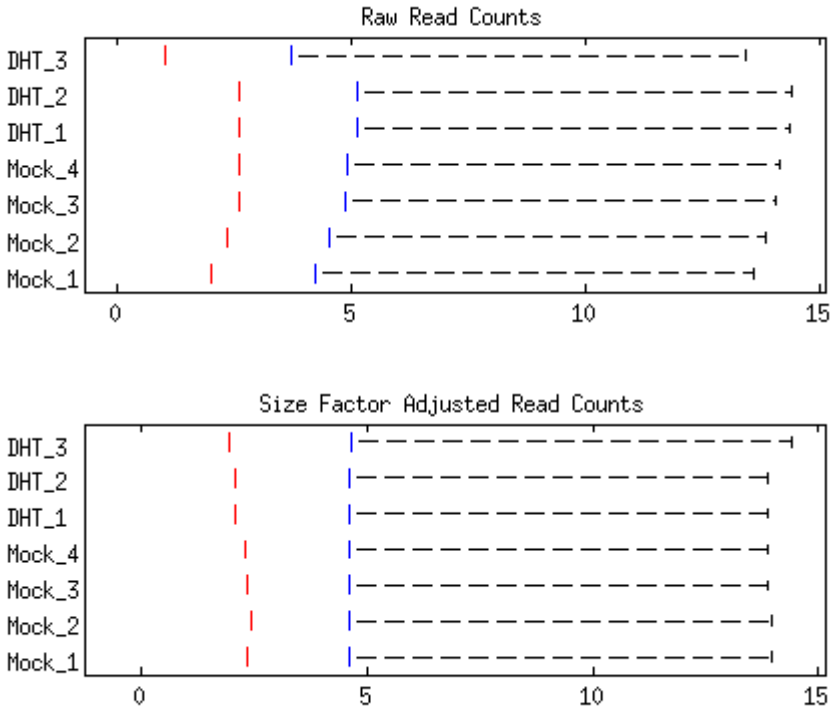


Figure 16: The box-plot is showing the count distribution of the mock-treated and DHT-treated samples and the size factor adjustment.

2.8 Estimating Negative Binomial Distribution Parameters

The counts values of a gene that we are expecting are also proportional to the gene abundance parameter. We estimated the gene abundance parameter of the counts average from samples corresponding to an experimental condition. We computed the counts mean and samples variance of the mock-treated samples.

```
base_mean_mock = mean(base_counts(:, cond_Mock), 2);
base_var_mock = var(base_counts(:, cond_Mock), 0, 2);
```

We used `estimateBaseParams` function to avoid code duplication in this experiment for computing parameters for samples from different conditions. This function computes the mean, the variance, and the diagnostic variance residual distribution from replicates under the same condition. For example, we computed the base means and variances for DHT-treated samples.

```
[base_mean_dht, base_var_dht] = estimateBaseParams(lncap_counts(:,
cond_DHT),...
sizeFactors(cond_DHT),...
'MeanAndVar');
```

In this model, the full variances of the Negative Binomial distribution of the counts of a gene, are considered as a sum of a shut noise term and raw variance term. The shut noise term means the reads counts of the gene, the raw variance can be predicted from the mean, i.e. genes with a similar expression level have similar variance across the replicates (samples of the same biological condition). The smooth function models the dependence of the raw variance on the mean and it is obtained by fitting the sample mean and variance within replicates for each gene using the local regression function `malowess`. We got the smooth fit data from the sample mean and variance of the mock-treated samples.

```
[rawVarSmooth_X_mock, rawVarSmooth_Y_mock] = ...
                                    estimateBaseParams(lncap_counts(:,
cond_Mock),...
sizeFactors(cond_Mock),...
'SmoothFunc');
```

After that, we found the raw variances for each gene from its base mean value by interpolation:

```
raw_var_mock_fit = interp1(rawVarSmooth_X_mock,
rawVarSmooth_Y_mock,...
                           log(base_mean_mock), 'linear', 0);
```

We added the bias correction term to get the raw variances:

```
zConst = sum(1 ./sizeFactors(cond_Mock), 2) /
length(sizeFactors(cond_Mock));
raw_var_mock = raw_var_mock_fit - base_mean_mock * zConst;
```

In the end, we were able to plot the sample variance and the raw variance data to check the fit of the variance function:

```
[base_mean_mock_sort, sidx] = sort(log10(base_mean_mock));
raw_var_mock_sort = log10(raw_var_mock_fit(sidx));

figure
plot(log10(base_mean_mock), log10(base_var_mock), '*')
hold on
line(base_mean_mock_sort, real(raw_var_mock_sort), 'Color', 'r',
'LineWidth',2)
ylabel('log10(base variances) of mock-treated samples')
xlabel('log10(base means) of mock-treated samples')
```

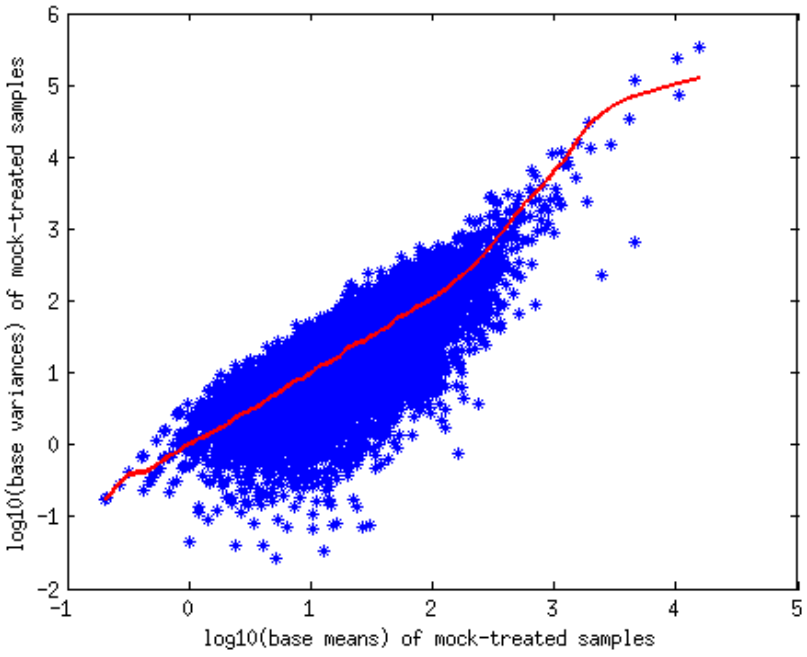


Figure 17: Sample variance and the raw variance data of the mock-treated samples

The red line in Figure 17 represents the fit, it follows well the single-gene estimates, even though the spread of the latter is considerable given that each raw variance value is estimated from only four values (four mock-treated replicates).

RNA-seq experiments have usually few replicates, sometimes the single-gene assessment of the base variance deviates wildly from the fitted value. We calculated the cumulative probability for the ratio of single-gene assessment of the base variance to the fitted value from the chi-square distribution [33] to see if the deviation is too wild.

We computed the cumulative probabilities of the variance ratios of mock/treated samples.

```
df_mock = sum(cond_Mock) - 1;
varRatio_mock = base_var_mock ./ raw_var_mock_fit;
pchisq_mock = chi2cdf(df_mock * varRatio_mock, df_mock);
```

Then we computed the empirical cumulative density functions (ECDF) stratified by base count levels and plotted the ECDFs curves. We divided the counts into seven levels.

```
count_levels = [0 3; 3.1 12; 12.1 30; 30.1 65; 65.1 130; 130.1 310;
310.1 2500];
figure;
hold on
cm = jet(7);
for i = 1:7
   [Y1,X1] = ecdf(pchisq_mock(base_mean_mock>count_levels(i, 1) &...
                              base_mean_mock<count_levels(i,2)));
   plot(X1,Y1, 'LineWidth',2,'color',cm(i,:))
end
plot([0,1],[0,1] , 'k', 'linewidth', 2)
set(gca, 'Box', 'on')
legend('0-3', '3-12', '12-30', '31-65', '65-130', '131-310', '311-
2500',...
      'Location','NorthWest')
xlabel('Chi-squared probability of residual')
ylabel('ECDF')
title('Residuals ECDF plot for mock-treated samples')
```

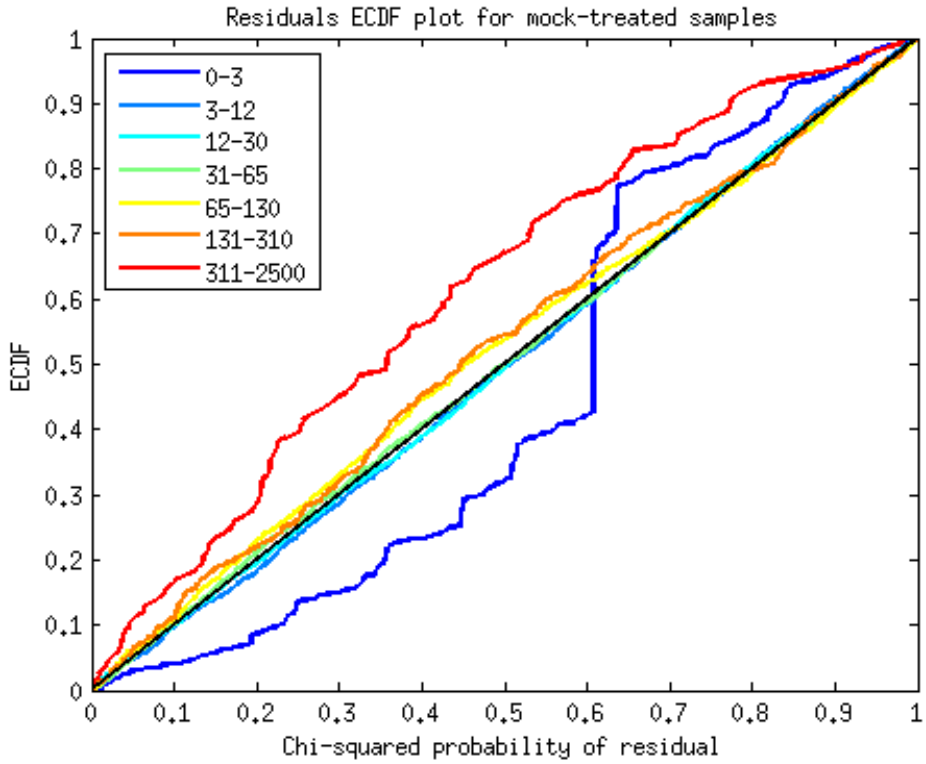


Figure 18: ECDF curves for mock-treated samples

In Figure 18 we can see that the curves of counts levels greater than 3 and below 130 follow the diagonal (black line) well. If the ECDF curves are below the diagonal, the variance is underestimated. If the ECDF curves are above the diagonal, the variance is overestimated [33]. For very low counts (below 3), the deviations becomes much more stronger, but at these levels dominates the shot noise. For the high counts (above 311), the variance is overestimated. It might be because there are not enough genes with high counts. We computed the number of genes in each of the counts levels.

```
num_in_count_levels = zeros(1, 7);
for i = 1:7
    num_in_count_levels(i) = sum(base_mean_mock>count_levels(i, 1) &
...
                                 base_mean_mock<count_levels(i,2));
end
num_in_count_levels
num_in_count_levels =

        4045      3365      3549      2493      1219       435       116
```

Increasing the sequence depth, which in turn increases the number of genes with higher counts, improves the variance estimation.

We produced the same ECDF plot for the DHT-treated samples.

```
pchisq_dht = estimateBaseParams(lncap_counts(:, cond_DHT),...
                                                    sizeFactors(1,
cond_DHT),...
                                                   'Diagnostic');
figure;
hold on
for i = 1:7
   [Y1,X1] = ecdf(pchisq_dht(base_mean_dht>count_levels(i, 1) & ...
                              base_mean_dht<count_levels(i,2)));
   plot(X1,Y1, 'LineWidth',2,'color',cm(i,:))
end
plot([0,1],[0,1] , 'k', 'linewidth', 2)
set(gca, 'Box', 'on')
legend('0-3', '3-12', '12-30', '31-65', '65-130', '131-310', '311-
2500',...
      'Location','NorthWest')
xlabel('Chi-squared probability of residual')
ylabel('ECDF')
title('Residuals ECDF plot for DHT-treated samples')
```

We can see the ECDF plot in Figure 19. In both cases, mock-treated and DHT-treated samples, most of the ECDF curves follow well the diagonal. The fits are soundly good.

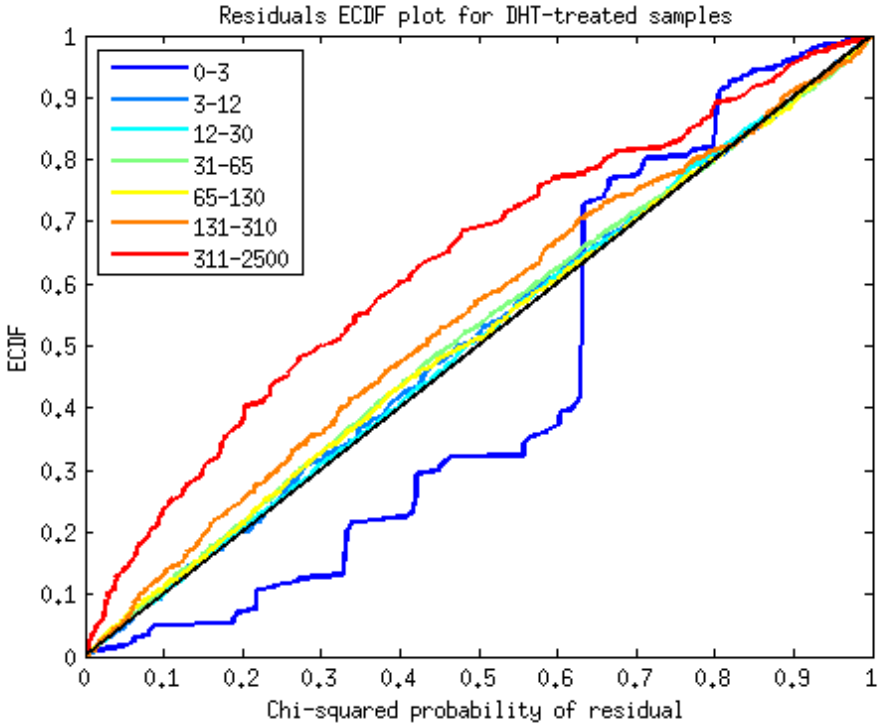


Figure 19: ECDF curves for DHT-treated samples

2.9 Testing for Differential Expression

After we estimated and verified the mean-variance dependence, we can start to test for differentially expressed genes between the samples in different biological conditions – mock-treated and DHT-treated. We used the function `estimateNBParams` to estimate the mean and full variance of the Negative Binomial distribution with two parameters for each gene from the three sets of parameters mentioned above.

```
[mu_mock, full_var_mock, mu_dht, full_var_dht] =...
          estimateNBParams(lncap_counts, sizeFactors, cond_DHT, cond_Mock);
```

Then we computed the p-values for the statistical significance of the change from DHT-treated condition to mock-treated condition. We used the function `computePVal` for implementing the numerical computation of the p-values presented in the reference [33]. We used another function `nbinpdf` to compute the Negative Binomial probability density.

We got the genes counts for each condition:

```
k_mock = sum(lncap_counts(:, cond_Mock), 2);
k_dht = sum(lncap_counts(:, cond_DHT), 2);
pvals =  computePVal(k_dht, mu_dht, full_var_dht, k_mock, mu_mock,
full_var_mock);
```

We adjusted the p-values from the multiple tests for false discovery rate (FDR) with the Benjamini-Hochberg [34] procedure using the `mafdr` function:

```
p_fdr = mafdr(pvals, 'BHFDR', true);
```

We determined the fold change estimated from the DHT-treated to the mock-treated condition:

```
foldChange = base_mean_dht ./ base_mean_mock;
```

We determined the base 2 logarithm of the fold change:

```
log2FoldChange = log2(foldChange);
```

We determined the mean expression level estimated from both conditions:

```
base_mean_com = estimateBaseParams(lncap_counts, sizeFactors,
'MeanAndVar');
```

And assumed a p-value cutoff of 0.01:

```
de_idx = p_fdr < 0.01;
```

Now we were able to plot the log2 fold changes against the base means and color those genes with p-values less than the cutoff value red:

```
figure;
plot(log2(base_mean_com(~de_idx, :)), log2FoldChange(~de_idx,:), 'b.')
hold on
plot(log2(base_mean_com(de_idx, :)), log2FoldChange(de_idx, :), 'r.')
xlabel('log2 Mean')
ylabel('log2 Fold Change')
```

We can see the result in Figure 20.

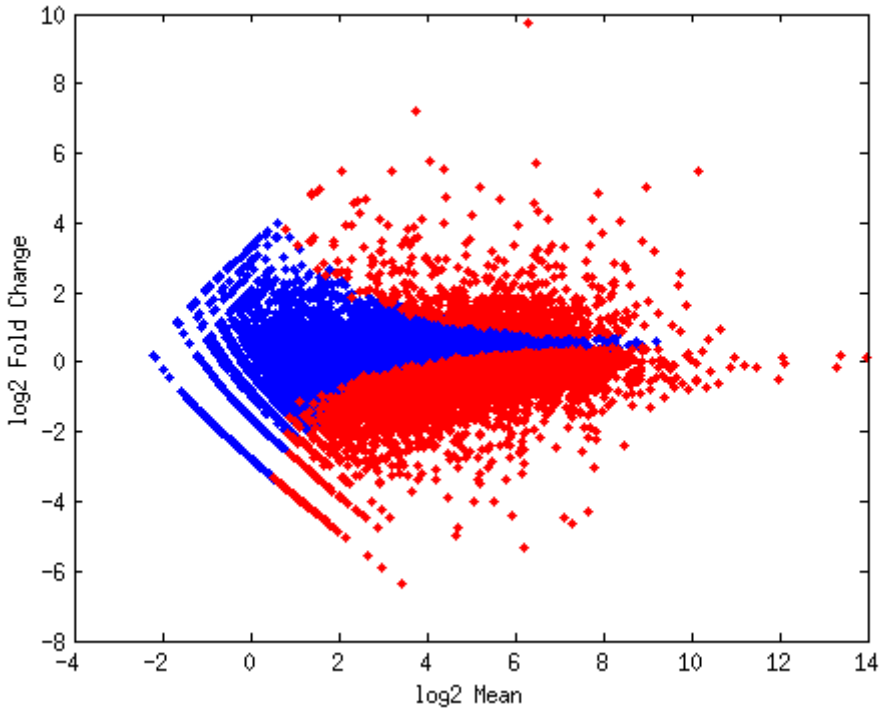


***Figure 20:** Plot of the log2 fold changes against the base means*

In the end, we could identify up- or down-regulated genes for the base mean of the counts levels over 3:

```
up_idx = find(p_fdr < 0.01 & log2FoldChange >= 2 & base_mean_com > 3 );
numel(up_idx)
ans = 178

down_idx = find(p_fdr < 0.01 & log2FoldChange <= -2 & base_mean_com > 3
);
numel(down_idx)
ans = 284
```

This analysis identified 462 genes (out of 20 327 genes) that were differentially up-or down-regulated by hormone treatment. 178 genes were up-regulated and 284 genes were down-regulated.

DISCUSSION

In this master thesis, we used a statistical test for assessing differential gene expression using RNA sequencing data. The aim was to figure out how androgen, DHT affects gene expression of the prostate cancer cells.

The analysis is based on parameters of the Negative Binomial distribution. It is extension of the Poisson model where the variance is larger than the mean. Another distribution, such as binomial or Poisson is not recommended for model the count variability in RNA-Seq data because of the overdispersion [48].

Important task before the analysis was to align the reads to human genome version hg19, GRCh37 using a B*owtie* aligner on Linux. The aligner was instructed to report one of the best valid alignment and no more than two mismatches were allowed for alignment. Next, the annotation object of target genes was created using Ensembl´s BioMart service and MATLAB's functions. We got the structure of protein coding genes including chromosome name, gene name, gene start/end, and strand direction. After importing mapped short read alignment data into MATLAB creating a BioMap and determining digital gene expression, we estimated the library size factor. The size factor was successfully adjusted to be the same for mock-treated and DHT-treated samples.

The Negative Binomial model parameters were estimated using functions included in statistical toolbox, whereas the distribution parameters of the single-gene were estimated well. We tested the differential gene expression between the mock and DHT-treated samples after estimating and verifying the mean-variance dependence. The high density of the mapped reads to annotated genes allows both qualitative and quantitative measurement of the transcription in response to the hormone treatment. To determine DHT-regulated genes in our prostate cancer model, we enumerated the number of the mapped reads to exons in individual transcripts before and after DHT stimulation. We compared the number of the reads mapped to specific transcripts to the total number of the reads mapped to all other transcripts to identify the DHT-regulated genes. We identified 462 genes that were differentially up- or down-regulated by the hormone treatment. Figure 20 shows the scatter plot of the gene expression in mock-treated and DHT-treated samples, differentially expressed genes were labeled red based on $p<0.01$. Qualitative analysis shows that 178 genes were up-regulated and 284 genes were down-regulated by DHT-treatment.

CONCLUSION

The main aim of this master thesis was to create a theoretical basis dealing with RNA-sequencing and next-generation sequencing and carry out the differential gene expression using a Negative Binomial model.

The first chapter discusses DNA sequencing in general including its importance. The next chapter focuses on RNA-sequencing, which provides a view of the whole transcriptome. Several RNA-sequencing methods were described, such as mRNA-Seq, Total RNA-Seq or Paired- End RNA-Seq. One of the important benefits of RNA-Seq is the possibility to capture all changes in gene expression.

The following chapter introduces the Next-Generation Sequencing, the revolutionary technology, that allows scientists to study the differential gene expression much more faster and cheaper than the original Sanger method. The basic principles of the NGS methods were described such as: 454 (Roche), Illumina, SOLIiD System (Applied Bioscience), Ion Torrent (Life Technologies) and Oxford Nanopore (Oxford NANAPORE Technologies). The systems have their characteristics, they vary in the length of read, the speed, the need of amplification or the price, that should be considered during the selection of the right system for a given project.

The prostate cancer data set was chosen and downloaded as a studied interested set of genes. The reads had to be mapped to the whole human genome version GRCh37 before the analysis that was later realized using MATLAB (version R2013b). *Bowtie* mapper based on Burrows-Wheeler transform was used for this task, it was proved as very fast short reads aligner, and it took few minutes on command line under Linux. The aligned reads in SAM format were then sorted using SAMtools, another useful tool for manipulation with SAM and BAM-formatted files.

The next task of this thesis was to create an annotation of target genes. The TSV table with all protein encoding genes was downloaded using Ensembl´s BioMart Service. The annotation of target genes was created in MATLAB environment as well as all the other tasks. The mapped short reads were imported into MATLAB creating a BioMap. After that, the created BioMap objects were used for determining digital gene expression, where the bioinformatics toolbox and its function was useful for this task. In the end, we estimated the library size factor and the Negative Binomial distribution parameters for the inference of the differential signal in RNA expression. The results of the gene expression were available after testing the data for differential gene expression. As output of this analysis, 462 genes (out of 20 327 genes) were

differentially up-or down-regulated by hormone treatment. 178 genes were up-regulated and 284 genes were down-regulated.

Differential gene expression is the most recent topic. Discoveries in differential gene expression may have significant outcomes of deep impact for the whole society. In my opinion, the differential gene expression will be more and more important in the research, where the scientists are inventing new methods of biological treatment. They need to know how the biological or chemical substance affects the gene expression. It could bring considerable progress in treatment especially of cancer´s diseases, if the doctors could change the gene expression in the cells.

REFERENCES

[1] HALIMAA, Pauliina. *NEXT GENERATION SEQUENCING*. University of Eastern Finland, 2013

[2] SONESON, Ch., DELORENZI, M. "*A Comparison of Methods for Differential Expression Analysis of RNA-seq Data*". BMC Bioinformatics; 14:91. DOI: 10.1186/1471-2105- 14-91, 2013 Mar 9.

[3] Illumina, Inc. ILLUMINA, Inc. *www.illumina.com* [online]. 2013 [cit. 2013-11-24].

[4] BASHIR, Ali, et al. *Evaluation of paired-end sequencing strategies for detection of genome rearrangements in cancer*. PLoS computational biology, 2008, 4.4: e1000051.

[5] ILLUMINA, Inc. *An Introduction to Next-Generation Sequencing Technology* [online]. Pub No. 770-2012-008. 2013 [cit. 2013-11-24].

[6] SCHUSTER, Stephan C. *Next-generation sequencing transforms today's biology*. Nature, 2007, 200.8.

[7] REIS-FILHO, Jorge S., et al. *Next-generation sequencing*. Breast Cancer Res, 2009, 11.Suppl 3: S12.

[8] ROTHBERG, Jonathan M.; LEAMON, John H. *The development and impact of 454 sequencing.* Nature biotechnology, 2008, 26.10: 1117-1124.

[9] ANSORGE, Wilhelm J. *Next-generation DNA sequencing techniques*. New biotechnology, 2009, 25.4: 195-203.

[10] MOROZOVA, Olena; MARRA, Marco A. *Applications of next-generation sequencing technologies in functional genomics. Genomics*, 2008, 92.5: 255-264.

[11] *New Sequencing Technologies and Their Clinical Impact.* In: SMITH, David. I. www.mayomedicallaboratories.com [online]. May 2010 [cit. 2013-11-24].

[12] © 2013 LIFE TECHNOLOGIES CORPORATION. Life technologies [online]. [cit. 2013-11-28].

[13] CHECK HAYDEN, Erika. *Nanopore genome sequencer makes its debut: Technique promises it will produce a human genome in 15 minutes.* Nanopore genome sequencer makes its debut. 2012, č. 10. DOI: 10.1038/nature.2012.10051.

[14] SCHADT, Eric E.; TURNER, Steve; KASARSKIS, Andrew. *A window into third-generation sequencing.* Human molecular genetics, 2010, 19.R2: R227-R240.

[15] GLENN, Travis C. *Field guide to next-generation DNA sequencers*. Molecular Ecology Resources, 2011, 11.5: 759-769.

[16] BLANDE, Daniel. *RNASeq Data Analysis Workflow*. University of Eastern Finland, 2013.

[17] SAWICKI, Mark P., et al. *Human genome project.* The American journal of surgery, 1993, 165.2: 258-264.

[18] COCK, Peter JA, et al. *The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants.* Nucleic acids research, 2010, 38.6: 1767-1771.

[19] EWING, Brent; GREEN, Phil. *Base-calling of automated sequencer traces using Phred. II. error probabilities.* Genome research, 1998, 8.3: 186-194.

[20] JONES, Siân, et al. *Core signaling pathways in human pancreatic cancers revealed by global genomic analyses.* Science, 2008, 321.5897: 1801-1806.

[21] PARSONS, D. WILLIAMS, et al. *An integrated genomic analysis of human glioblastoma multiforme. Science*, 2008, 321.5897: 1807-1812.

[22] SHIMIZU, Masahito, et al., *Epigallocatechin gallate and polyphenon E inhibit growth and activation of the epidermal growth factor receptor and human epidermal growth factor receptor-2 signaling pathways in human colon cancer cells.* Clinical Cancer Research, 2005, 11.7: 2735-2746.

[23] BASELGA, J. The EGFR as a target for anticancer therapy—focus on cetuximab. *European Journal of Cancer*, 2001, 37: 16-22.

[24] LI, H., LOVCI, M. T., KWON, Y-S., ROSENFELD, M. G., FU, X-D., and YEO, G. W. "*Determination of Tag Density Required for Digital Transcriptome Analysis: Application to an Androgen-Sensitive Prostate Cancer Model"*, PNAS, 105(51), pp 20179-20184, 2008.

[25] LI, H., et al. *The sequence alignment/map format and SAMtools*. Bioinformatics, 2009, 25.16: 2078-2079.

[26] THE SAM/BAM FORMAT SPECIFICATION WORKING GROUP. *Sequence Alignment/Map Format Specification*. 29 May 2013.

[27] LANGMEAD, Ben, et al. *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.* Genome Biol, 2009, 10.3: R25.

[28] Ensemble release 74. *About the Ensembl Project* [online]. December 2013 © [cit. 2013-12-08]. Available at: http://www.ensembl.org/info/about/index.html

[29] Ensemble release 74. *Ensembl Genome Browser* [online]. December 2013 © [cit. 2013-12-08]. Available at: http://www.ensembl.org/info/data/biomart.html

[30] UCSC Genome Browser: Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. *Genome Res.* 2002, Jun;12(6):996-1006.

[31] MORTAZAVI, A., WILLIAMS, B.A., McCUE, K., SCHAFFER, L., and WOLD, B. *Mapping and quantifying mammalian transcriptomes by RNA-Seq*, Nature Methods, 5, pp 621-628, 2008.

[32] ROBINSON, M.D., OSHLACK, A. "*A Scaling Normalization method for differential Expression Analysis of RNA-seq Data*", Genome Biology 11:R25, 1-9, 2010.

[33] ANDERS, S. HUBER, W. "*Differential Expression Analysis for Sequence Count Data*", Genome Biology, 11:R106, 2010.

[34] BENJAMINI, Y., HOCHBERG, Y. "*Controlling the false discovery rate: a practical and powerful approach to multiple testing*", J. Royal Stat. Soc., B 57, 289-300, 1995.

[35] "*ACS :: What Is Prostate Cancer*?" American Cancer Society :: Information and Resources for Cancer: Breast, Colon, Prostate, Lung and Other Forms. Web. 15 June 2010.

[36] "*IARC Worldwide Cancer Incidence Statistics—Prostate*". JNCI Cancer Spectrum. Oxford University Press. December 19, 2001. Archived from the originalon February 5, 2006.

[37] MADU, C., LU, Y. *Novel diagnostic biomarkers for prostate cancer*. J Cancer 2010; 1:150-177. doi:10.7150/jca.1.150.

[38] MILLER, DC, HAFES, KS, STEWART, A, MONTIE, JE, WEI, JT (September 2003). "*Prostate carcinoma presentation, diagnosis, and staging: an update form the National Cancer Data Base*". Cancer 98 (6): 1169-78. doi:10.1002/cncr.11635.PMID 12973840.

[39] VAN DER CRUIJSEN-KOETER, IW, VIS, AN, ROOBOL, et.al. (July 2005). "*Comparison of screen detected and clinically diagnosed prostate cancer in the European randomized study of screening for prostate cancer, section rotterdam*". Urol 174 (1): 121–5.doi:10.1097/01.ju.0000162061.40533.0f.

[40] HSING, AW, CHOKKALINGAM, AP (2006). "*Prostate cancer epidemiology*". Frontiers in Bioscience 11: 1388–413. doi:10.2741/1891.

[41] HANKEY, BF, FEUER, EJ et. al, (June 16, 1999). "*Cancer surveillance series: interpreting trends in prostate cancer-part I: Evidence of the effects of screening in recent prostate cancer incidence, mortality, and survival rates*". J Natl Cancer Inst 91 (12): 1017–24. doi:10.1093/jnci/91.12.1017. PMID 10379964.

[42] MARTIN, RM, VATTEN, L, GUNNELL, D, ROMUNDSTAD, P (March 2010). "*Blood pressure and risk of prostate cancer: cohort Norway (CONOR)"*. Cancer Causes Control 21(3): 463–72. doi:10.1007/s10552-009-9477-x. PMID 19949849.

[43] BONEKAMP, D, JACOBS, MA, et. al. (May-June 2011). *"Advancements in MR Imaging of the Prostate: From Diagnosis to Interventions"*. Radiographics 31:677-703. doi:10.1148/rg.313105139. PMC 3093638. PMID 21571651.

[44] WARMKESSEL, J. (2006). *Contemporary issues in prostate cancer: a nursing perspective*. Jones & Bartlett Learning. pp. 108–. ISBN 978-0-7637-3075-8

[45] MONGIAT-ARTUS, P, PEYROMAURE, M, et. al. (December 2009). *"Recommendations for the treatment of prostate cancer in the elderly man: A study by the oncology committee of the French association of urology". Prog. Urol.* **19** (11): 810–7. doi:10.1016/j.purol.2009.02.008.

[46] LYCKEN, M. et al., *Patterns of androgen deprivation therapies among men diagnosed with localised prostate cancer: A population-based study*, Eur J Cancer (2014), http://dx.doi.org/10.1016/j.ejca.2014.03.279.

[47] HO, J., et. al., *Differential variability analysis of gene expression and its application to human diseases, Bioinformatics* (2008) 24 (13):i390-i398.

[48] DI, Y., SCHAFER, D. W., CUMBIE, J. S., CHANG, J. H. *The NBP Negative Binomial Model for Assessing Differential Gene Expression from RNA-Seq* . Statistical Applications in Genetics and Molecular Biology, Volume 10, Issue 1, Pages 1–28, ISSN (Online) 1544-6115, 2011.

LIST OF IMAGES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADT	-	Androgen Deprivation Therapy
BAM	-	binary version of SAM
BGZF	-	Blocked GNU Zip Format
bp	-	base pair
BWT	-	Burrows-Wheeler Transform
cDNA	-	complementary DNA
DHT	-	dihydrotestosterone
DNA	-	deoxyribonucleic acid
ECDP	-	Empirical Cumulative Density Functions
FDR	-	False Discovery Rate
gDNA	-	genomic DNA
GFF	-	General Feature Format
LNCaP cells	-	androgen-sensitive human prostate adenocarcinoma cells
MATLAB	-	Matrix Laboratory
MRI	-	Magnetic Resonance Imaging
mRNA	-	messenger RNA
NCBI	-	National Center for Biotechnology Information
NGS	-	Next-generation sequencing
PGM	-	Personal Genome Machine
PSA	-	Prostate Specific Antigen
RNA	-	ribonucleic acid
RNA-Seq	-	Sequencing of RNA
rRNA	-	ribosomal RNA
SAM	-	Sequence Alignment/Map
SOLiD	-	Supported Oligonucleotide Ligation and Detection system
TSV	-	tab-separated-value

SUPPLEMENT – MATLAB CODE

```
% Master thesis:
% Differential Gene Expression using a negative binomial model
% Tereza Janakova
% 2014

% creating bowtie indexes
bowtiebuild('hg19.fas', 'hg19')


% mapping the reads to the human genome using Bowtie mapper

% Ordered the SAM-formatted files by reference name first, then by
genomic position using SAMtools.

%% Creating an annotation object of target genes

GFFfilename = ensemblmart2gff('ensemblmart_genes_hum37.txt');
genes = GFFAnnotation(GFFfilename)

% Create a subset with the genes present in chromosomes only.
chrs =
{'1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','
16','17','18','19','20','21','22','X','Y','MT'};
genes = getSubset(genes,'reference',chrs)

%% Importing Mapped Short Read Alignment Data

bm1 = BioMap('s1.sort.sam')
getSummary(bm1)

%% Determining Digital Gene Expression

 geneReference =  seqmatch(genes.Reference,chrs,'exact',true);

 counts1 =
getCounts(bm1,genes.Start,genes.Stop,1:genes.NumEntries,geneReferenc
e);

filenames =
{'s1.sort.sam','s2.sort.sam','s3.sort.sam','s4.sort.sam','s5.sort.sa
m','s6.sort.sam','s8.sort.sam'};
samples =
{'Mock_1','Mock_2','Mock_3','Mock_4','DHT_1','DHT_2','DHT_3'};

lncap_counts =
bioma.data.DataMatrix(NaN([genes.NumEntries,7]),genes.Feature,sample
s)
lncap_counts(:,1) = counts1

lncap_counts(190:200,:)

lichr1 = geneReference == 1;% logical index to genes in chromosome 1
sum(lncap_counts(:,1) >= 50 & lichr1)
```

```
% Repeat this step for the other six samples (SAM files) in the data
set to get their gene counts and copy the information to the
previously createdÂ DataMatrix.
for i = 2:7
    bm = BioMap(filenames{i});
    counts =
getCounts(bm,genes.Start,genes.Stop,1:genes.NumEntries,geneReference
);
    lncap_counts(:,i) = counts;
end

% Inspect the first 10 rows in the count table.
lncap_counts(190:200, :)

% The DataMatrixÂ lncap_countsÂ contains counts for samples from two
biological conditions: mock-treated and DHT-treated.
cond_Mock = logical([1 1 1 1 0 0 0]);
cond_DHT  = logical([0 0 0 0 1 1 1]);

ichr1 = find(lichr1);  % linear index to genes in chromosome 1
[~,h] = sort(genes.Start(ichr1));
ichr1 = ichr1(h);      % linear index to genes in chromosome 1
                       % sorted by genomic position

figure
plot(genes.Start(ichr1), lncap_counts(ichr1,'Mock_1'), '.-r',...
     genes.Start(ichr1), lncap_counts(ichr1,'DHT_1'), '.-b');
ylabel('Gene Counts')
title('Gene Counts on Chromosome 1')
fixGenomicPositionLabels(gca)  % formats tick labels and adds
datacursors
chromosomeplot('hs_cytoBand.txt', 1, 'AddToPlot', gca)

%% Inference of Differential Signal in RNA Expression
% 1. Library size parameters;
% 2. Gene abundance parameters under each experimental condition;
% 3. The smooth functions that model the dependence of the raw
variance on the expected mean.

% 1. Estimating Library size factor
geoMeans = exp(mean(log(lncap_counts), 2));

ratios = dmbsxfun(@rdivide, lncap_counts(geoMeans >0, :),
geoMeans(geoMeans >0));
sizeFactors = median(ratios, 1);

base_counts = dmbsxfun(@rdivide, lncap_counts, sizeFactors);

figure
subplot(2,1,1)
maboxplot(log2(lncap_counts), 'title','Raw Read Counts',...
                              'orientation', 'horizontal')
subplot(2,1,2)
maboxplot(log2(base_counts), 'title','Size Factor Adjusted Read
Counts',...
                             'orientation', 'horizontal')
```

```
% Estimating Negative Binomial Distribution Parameters

base_mean_mock = mean(base_counts(:, cond_Mock), 2);
base_var_mock = var(base_counts(:, cond_Mock), 0, 2);

[base_mean_dht, base_var_dht] = estimateBaseParams(lncap_counts(:,
cond_DHT),...

sizeFactors(cond_DHT),...
                                                    'MeanAndVar');

[rawVarSmooth_X_mock, rawVarSmooth_Y_mock] = ...
                                  estimateBaseParams(lncap_counts(:,
cond_Mock),...

sizeFactors(cond_Mock),...
                                                    'SmoothFunc');
% Find the raw variances for each gene from its base mean value by
interpolation.
raw_var_mock_fit = interp1(rawVarSmooth_X_mock,
rawVarSmooth_Y_mock,...
                             log(base_mean_mock), 'linear', 0);

% ...Add the bias correction term to get the raw variances
zConst = sum(1 ./sizeFactors(cond_Mock), 2) /
length(sizeFactors(cond_Mock));
raw_var_mock = raw_var_mock_fit - base_mean_mock * zConst;

% Plot the sample variance and the raw variance data to check the
fit of the variance function.
[base_mean_mock_sort, sidx] = sort(log10(base_mean_mock));
raw_var_mock_sort = log10(raw_var_mock_fit(sidx));

figure
plot(log10(base_mean_mock), log10(base_var_mock), '*')
hold on
line(base_mean_mock_sort, real(raw_var_mock_sort), 'Color', 'r',
'LineWidth',2)
ylabel('log10(base variances) of mock-treated samples')
xlabel('log10(base means) of mock-treated samples')

% Compute the cumulative probabilities of the variance ratios of
mock-treated samples.
df_mock = sum(cond_Mock) - 1;
varRatio_mock = base_var_mock ./ raw_var_mock_fit;
pchisq_mock = chi2cdf(df_mock * varRatio_mock, df_mock);

% Compute the empirical cumulative density functions (ECDF)
stratified by base count levels, and show the ECDFs curves. Group
the counts into seven levels.
count_levels = [0 3; 3.1 12; 12.1 30; 30.1 65; 65.1 130; 130.1 310;
310.1 2500];
figure;
hold on
cm = jet(7);
for i = 1:7
```

```
   [Y1,X1] = ecdf(pchisq_mock(base_mean_mock>count_levels(i, 1) &...
                              base_mean_mock<count_levels(i,2)));
   plot(X1,Y1,'LineWidth',2,'color',cm(i,:))
end
plot([0,1],[0,1] ,'k', 'linewidth', 2)
set(gca, 'Box', 'on')
legend('0-3', '3-12', '12-30', '31-65', '65-130', '131-310', '311-
2500',...
       'Location','NorthWest')
xlabel('Chi-squared probability of residual')
ylabel('ECDF')
title('Residuals ECDF plot for mock-treated samples')

num_in_count_levels = zeros(1, 7);
for i = 1:7
    num_in_count_levels(i) = sum(base_mean_mock>count_levels(i, 1) &
...
                                 base_mean_mock<count_levels(i,2));
end
num_in_count_levels

% produce the same ECDF plot for DHT-treated samples
pchisq_dht = estimateBaseParams(lncap_counts(:, cond_DHT),...
                                                 sizeFactors(1,
cond_DHT),...
                                                 'Diagnostic');
figure;
hold on
for i = 1:7
   [Y1,X1] = ecdf(pchisq_dht(base_mean_dht>count_levels(i, 1) & ...
                             base_mean_dht<count_levels(i,2)));
   plot(X1,Y1,'LineWidth',2,'color',cm(i,:))
end
plot([0,1],[0,1] ,'k', 'linewidth', 2)
set(gca, 'Box', 'on')
legend('0-3', '3-12', '12-30', '31-65', '65-130', '131-310', '311-
2500',...
       'Location','NorthWest')
xlabel('Chi-squared probability of residual')
ylabel('ECDF')
title('Residuals ECDF plot for DHT-treated samples')

%% Testing for Differential Expression

[mu_mock, full_var_mock, mu_dht, full_var_dht] =...
          estimateNBParams(lncap_counts, sizeFactors, cond_DHT,
cond_Mock);

k_mock = sum(lncap_counts(:, cond_Mock), 2);
k_dht = sum(lncap_counts(:, cond_DHT), 2);
pvals =  computePVal(k_dht, mu_dht, full_var_dht, k_mock, mu_mock,
full_var_mock);

p_fdr = mafdr(pvals, 'BHFDR', true);

foldChange = base_mean_dht ./ base_mean_mock;
```

```
log2FoldChange = log2(foldChange);

base_mean_com = estimateBaseParams(lncap_counts, sizeFactors,
'MeanAndVar');

de_idx = p_fdr < 0.01;

figure;
plot(log2(base_mean_com(~de_idx, :)), log2FoldChange(~de_idx,:),
'b.')
hold on
plot(log2(base_mean_com(de_idx, :)), log2FoldChange(de_idx, :),
'r.')
xlabel('log2 Mean')
ylabel('log2 Fold Change')

% identify up- or down- regulated genes for mean base count levels
over 3.

up_idx = find(p_fdr < 0.01 & log2FoldChange >= 2 & base_mean_com > 3
);
numel(up_idx)

down_idx = find(p_fdr < 0.01 & log2FoldChange <= -2 & base_mean_com
> 3 );
numel(down_idx)
```