Final Report

EECE399 - Field Experience in Computer Engineering

Summer 2013

Student name:	**Sharbel Dahlan**
Student number:	**1004018456**
Submitted to:	**Dr. Jinane Biri**
Date:	**August 27, 2013**

I pledge that this work is entirely my own.

Acknowledgements

Many thanks to Dr. Zaid Al-Ars for his continuous support and supervision throughout the entire field experience. I would also like to thank Hassan Sinno for sharing his experience and collaboration through the days and nights of work.

Contents

Acknowledgements i

List of Figures iii

List of Tables iv

Abbreviations v

1 Introduction 1

2 Workplace Setting 2

3 Field Experience Activities 3
- 3.1 Software Training . . . 3
- 3.2 Field Inspection . . . 4
- 3.3 Understanding the BLASR Aligner . . . 5
- 3.4 Analysis and Results . . . 6

4 Deliverables 8
- 4.1 Genetic Sequence Alignment . . . 8
- 4.2 The BLASR Aligner . . . 9

5 Conclusions 14

References 15

A Quality Control and Filtering of Reads (Part of the Alignment Process) 16
- A.1 Opening Read Files in FASTQ Format . . . 16
- A.2 Quality Evaluation using FastQC . . . 20
- A.3 Filtering using Trimmomatic . . . 28

B Running BLASR Programs 33
- B.1 Downloading BLASR . . . 33
- B.2 Running BLASR . . . 35

C Learning Outcomes (From Course Syllabus) 38

List of Figures

4.1 Summary of the alignment process 9
4.2 A sample output of the BLASR aligner with the "verbose" option 11
4.3 A segment of the verbose output of the BLASR aligner 12
4.4 Another segment of the verbose output 13

A.1 Accessing the CEN.PK113-7D sequence 17
A.2 The raw CEN.PK113-7D sequence file in .fastq format 18
A.3 The .fastq file annotated 19
A.4 The basic stats of the raw CEN.PK113-7D sequence. 21
A.5 The per base sequence quality of the raw sequence 22
A.6 The per sequence quality scores of the raw sequence 23
A.7 The per base sequence content of the raw sequence 24
A.8 The per base GC content of the sequence 25
A.9 The per sequence GC content of the sequence 26
A.10 The per base n content 27
A.11 The sequence length distribution 27
A.12 The sequence duplication level 28
A.13 Filtering the CEN.PK113-7D sequence using trimmomatic 29
A.14 The trimmed sequence 30
A.15 The basic stats of the filtered CEN.PK113-7D sequence (after trimming) . 31
A.16 The per base sequence quality of the trimmed CEN.PK113-7D sequence . . 32

B.1 BLASR folder and help menu 35
B.2 BLASR output in .fasta format 36
B.3 BLASR output in .bas.h5 format 37

List of Tables

3.1 Covered Learning Outcomes 3

A.1 Percentage of reads passing the filter after trimming 29

Abbreviations

BLASR	**B**asic **L**ocal **A**lignment (with) **S**uccessive **R**efining
CE (Group)	**C**omputer **E**ngineering (Group)
GPU	**G**raphics **P**rocessing **U**nit
NGS	**N**ext **G**eneration **S**equencing
SMS	**S**ingle **M**olecule **S**equencing
TU (Delft)	**T**echnische **U**niversiteit (Delft)
WGA	**W**hole **G**enome **A**lignment

Chapter 1

Introduction

The filed experience has taken place in the computer engineering lab of Delft University of Technology (TU Delft). Located in Delft, The Netherlands, the university is one of the leading technology and research universities in Europe. Several research topics take place, all of which are very recent, relating to the motto of the university "Challenge the future". The technical field in which the field experience was conducted was mainly bioinformatics. There has been ongoing research on improving bioinformatics algorithms (by parallelizing them) and implementing them on different supercomputing platforms to ensure high performance from both software and hardware. The improved algorithms by computer engineers can hopefully help the bioinformaticians to recognize flaws in genetic data. This in turn would help the molecular biologists (and hence physicians) to find better solutions to address various human diseases and health issues, including cancer. Since the field experience is in computer engineering, the focus will be more on the implementation of different bioinformatics algorithms and analysis of those algorithms. Nevertheless, rather than being limited to computer engineering, the knowledge and learning experience extends to other major fields like bioinformatics and molecular biology. Information about the nature of the work can be found in *Chapter 2: Workplace Setting*. Details on the activities performed and the relevant learning outcomes are found in *Chapter 3: Field Experience Activities*, while details on the material learned and some of the obtained output are found in *Chapter 4: Deliverables*. *Chapter 5: Conclusions* contains a summary of the lessons that were learned during the time of the field experience.

Chapter 2

Workplace Setting

Although the workplace was mainly based in the computer engineering lab of the CE group of the university, it was not restricted to one specific form for the whole period of the internship. At first, only individual work was done in the lab. This individual work included learning and gaining the background by reading a thesis paper previously done by a master student from TU Delft. In addition, practical work was also performed individually in the lab by running a series of programs related to the topic of the field experience. Later on, the workplace has switched to working in a team with another member from the American University of Beirut. With the setting also based in the CE lab, the work has included reading papers for recent projects, analyzing results, and running the programs that were used in order to obtain the results and enhance the algorithms. At all stages, the work was done under the supervision of Dr. Zaid Al-Ars, with whom frequent meetings were made to discuss the progress, ask questions, and share thoughts.

Chapter 3

Field Experience Activities

The activities performed in the field experience are grouped in the following sections of this chapter. Table 3.1 contains the numbers of the learning outcomes that are addressed in each field experience activity. The list of learning outcomes from the syllabus can be found in *Appendix C*.

TABLE 3.1: Covered Learning Outcomes

Activity	Section	Learning Outcomes Covered
Software Training	3.1	6, 7, and 8
Field Inspection	3.2	5, 6, and 8
Understanding the BLASR Aligner	3.3	4, 5, and 7
Analysis and Results	3.4	4, 5, 6, and 7

The upcoming sections of this chapter contain further explanation of how each learning outcome was covered.

3.1 Software Training

The software training that took place in the field experience was not only limited to getting familiar with the software and programs used. Instead, a large part of the training involved learning the background behind the project, which includes Biology, Mathematics, and Computer Science. Much of the information of this background was learned from a thesis paper done by a master student from TU Delft in 2010. The paper was about applying sequencing algorithms on the graphics processing unit (GPU). More information was learned while performing the practical training, which involved running a series programs

to sequence part of the genome of the yeast strain *saccharomyces cerevisiae*. Rather than just running the programs in the tutorial, a lot of information about the algorithms that the programs use was learned. Some extra work was done to understand the output of each of the programs and how it was obtained. Although the details about the programs were not included in the tutorial itself, understanding them was important in order to get the whole picture of how Mathematics (Graph Theory) and Computer Science (Algorithmics), together combined to represent Biological concepts, are related to create the programs used in the project. All the training work, although done individually, was very helpful in contributing to a better understanding of the project and its purpose. A summary of the information learned from the paper and the practical training, along with some results obtained, are in *Chapter 4: Deliverables* and *Appendix A*.

By performing those activities, many of the issues that were relevant to the bioinformatics industry were learned. The impact of the engineering activities carried, was understood to be related to human health, mainly targeting the issues of curing cancer. A better idea was formed on how the computer engineers are related to bioinformaticians, who are related to biologists, who in turn are related to physicians. Further work was yet to be done in order to better understand the issue of finding a cure for cancer. Moreover, learning individually doing extra research, which was a common factor in those activities, has contributed to the life-long learning and professional development during the internship.

3.2 Field Inspection

The inspection work has began by reading about the IBM Power 7 supercomputing platform and inspecting the architecture of the PowerLinux 7R2 server. The purpose of inspecting the Power 7 machine was to implement the algorithms that were learned (or yet to be learned) directly on those machines once they are readily installed. The server structure was inspected using a 3D animation from `http://ibmtvdemo.edgesuite.`

`net/servers/power/demos/powerlinux_7r2/index.html`. The animation enables to virtually view both the outer the inner structures of the server. Structures such as storage, processing unit, and memory were thoroughly inspected. However, the facility of physically working on the machines and applying the sequencing programs on them was not possible to be done at the time of the internship. This is because installation of the servers was delayed by the university technicians due to certain unknown reasons. Therefore, the focus was changed from learning about the machine to running alignment programs on a Linux system.

Although the time in which this activity was carried was little, it was sufficient to get an idea of how the successfully running the sequencing algorithms on the Power 7 machines impacts the health society. Having the sequencing algorithms applied on the machines would certainly be a forward step in the ongoing research on curing cancer.

3.3 Understanding the BLASR Aligner

The major activity done in the field experience was to run a sequence alignment program called BLASR, which was recently created. The activity subdivides into understanding how the aligner works and then running the program and analyzing the output (section 3.4). The creators of BLASR are researchers from University of California, San Diego, who have written a paper about the aligner, titled "Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory". The task to be done for this activity was to read the paper, which contains information about BLASR, and understand the algorithm used to create the alignment program. The paper also contains the data and results after running the program on sample genomes of two organisms: E. coli and H. sapiens. Many challenges were faced while performing this activity. One main challenge is the level of the information provided in the paper. There was a lot of effort spent on understanding the material and relating it to the material that is previously learned (during the training period). Although several aligning

algorithms were learned previously (some with similarities), the BLASR algorithm has its own characteristics to be understood. Another challenge is that, since the whole project is recent, there are not many resources yet to get more information or further explanation from. This led to make assumptions in order to keep going with the task. As an end result of this activity, a report summarizing the algorithm was successfully created. A brief description of the information concluded in the created report is found in *Chapter 4: Deliverables*.

In this activity, many aspects of the professional work were experienced. The work was done in a team with another member who is a senior computer engineering student from the American University of Beirut (AUB). Being in a team and collaborating on a project was very beneficial because learning would be improved as the experiences are shared. In addition during the process of getting the job done, documenting the steps, and writing reports to summarize the information, new ways and adjustments were developed on the way to do all that work efficiently. This in turn adds to the experience of life-long learning and professional development while doing the activities. Adding to that, ethical issues were taken care of, such as managing to have a separate descriptive report while preserving the correct information. Furthermore, holding meetings with the supervisor, raising concerns, flagging problems, and presenting the work, were all in favor of improving the communication skills, which is a vital part in the professional environment.

3.4 Analysis and Results

This activity involves installing the the BLASR aligner, running it, getting output data, and performing analysis to the data to be able to obtain useful results that are similar to those in the BLASR paper. The purpose of reproducing the results by TU Delft is to perform benchmarking with the results of the algorithms from University of San Diego. Once the BLASR programs are successfully run and similar results of the existing paper are obtained, TU Delft can begin to write another BLASR paper with the benchmarks.

While downloading the data and installing the program seemed quick tasks, they took more time than expected. Even downloading the prerequisite programs and libraries have taken some time (since their source was not directly given). The data were found in different file formats and it took a while to find out which format works for BLASR, or if not available, download programs to convert between the formats. At many points, several programs and toolkits that were assumed to be useful were downloaded. However, not all of them were used in obtaining the expected results. Throughout the process of running the BLASR aligner, documentation (which can be found in Appendix B) was performed to the steps in order to simplify later usage of the program. Once the program was finally run, the data was obtained. Nevertheless, while some of the obtained data was familiar due to previously performing the practical tutorial, most of the data was seen for the first time. Hence, meaning of the data was searched, as well as some assumptions were made. In an attempt to try and come up with results that match the data in the BLASR paper, several questions came across as the data obtained did not exactly match with the paper's. Thus, the creators of the BLASR aligner were contacted for certain inquiries on how the data obtained translates to the results shown in the BLASR paper. The communication with the researchers was necessary to get a better idea on the process followed to come up with the results, which was helpful in this case since the goal is to reproduce the results for the purpose of benchmarking.

Once more in this activity, teamwork was experienced to perform the task of successfully running the BLASR aligner. Aside from that, getting the program to run given the limited resources serves as an example to professional development activity. Equally, seeking professional help from the researchers when issues were encountered throughout the activity is an example that has improved the professional development. The outcome from this activity has a significant global impact, since coming up with effective algorithms aids the bioinformatics industry, which helps in finding the solution to several health issues related to genetics.

Chapter 4

Deliverables

4.1 Genetic Sequence Alignment

One of the main processes in which bioinformatics is involved is the alignment of genetic sequences, which is used in many applications, such as determining the origin of a sequence or constructing evolutionary trees . In the alignment process, a sequence of nucleotides (A, C, T, and G), which are the bases that consist the DNA, is lined up with another sequence in order to maximize the degree of similarity between the two [3]. Initially, a DNA sample from an organism is crushed into smaller segments, which are called reads. Those reads compose the raw sequence data, and can be represented by strings of the letters of the nucleotides (A,C, T, and G). Before processing those reads, quality inspection and filtering is performed to them. The reads are then filtered and assembled into structures called contigs, which are aligned into scaffolds. Then, gene prediction is performed to obtain annotated scaffolds. The annotated scaffolds are then mapped against a reference genome, which is an already existing aligned genome. The existence of the reference genome simplifies the alignment process. Mapping results in mapped reads, which are later analyzed by various analysis techniques. Figure 4.1 shows the alignment process summarized.

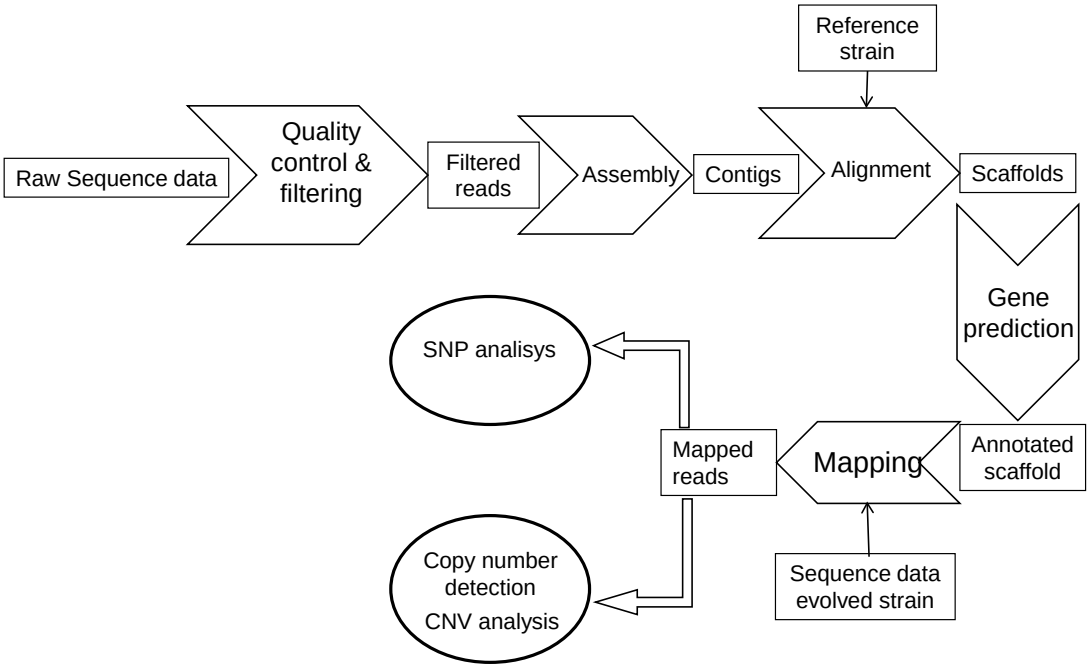

FIGURE 4.1: Summary of the alignment process

This process was experienced when performing the practical tutorial involving the alignment of a segment of the yeast strain *saccharomyces cerevisiae*. The quality analysis was performed using a program called FastQC, while the filtering of reads was performed using a program called trimmomatic. The quality control and filtering part of the alignment is shown in more details in *Appendix A* of this report.

4.2 The BLASR Aligner

At many certain cases, the size of the reads consists of thousands of bases (kilobases). Although Next-Generation Sequencing (NGS) techniques already exist, they are not very efficient in aligning long reads. As a solution to that, BLASR was created by Mark Chaisson and Glenn Tesler, from the University of California, San Diego. BLASR uses coarse alignment methods developed during Whole Genome Alignment (WGA) studies, and then makes use of the advanced data structures utilized in NGS mapping studies to

speed up the process [2]. Hence, it focuses on speed to perform Single Molecule Sequencing (SMS) reads with high error rates. Error rates are caused by insertion or deletion of bases. The SMS is performed as follows: First, cluster of short exact matches between the read and the genome it is mapped against is found, using either a suffix array or BWT-FM index. Then, a more detailed alignment is performed to the regions where reads are matched to assign the alignment. Two conditions should be met for BLASR to be feasible:

1. there should be a significant number of matches of minimal length between the read and the genome
2. there should not be a high number of regions where the read can map to with high identity (low number of false positive clusters)

One organism that meets the conditions is the human genome, and thus BLASR can be applied to the human genome, as what was done. The BLASR aligner was run by aligning reads from the human genome to a reference of the Escherichia Coli Bacterium.

Appendix B contains detailed information on installing BLASR programs and running them on a linux system. When the program was run, different outputs were obtained. Some analysis was done to those outputs based on the information learned as well as on some assumptions according to the nature of the data. There are several options that are used while running the blasr command to output the files, and each option concentrates on one aspect of the output. In this section, only one sample output is shown (which is obtained when using the "verbose" option). Since the read files are large in size, it is impossible o show the whole output, and therefore segments of the output are shown in Figures 4.2 to 4.4.

FIGURE 4.2: A sample output of the BLASR aligner showing the reads that were aligned to the reference genome. The option chosen while running BALSR to obtain this output is the "verbose" option, which shows detailed visuals of the result of the alignment. The numbers beside each sequence represent the positions of the bases in each genome. Visually shown are the matches, the mismatches, and the gaps.

```
aligning interval : 598 809247 809264 306 323 809247 to 809264 1.3
interval align score: -85
         Query: SRR305922.5 length=598/0_598
        Target: TY-2482_chromosome
         Model: a hybrid of global/local non-affine alignment
     Raw score: -85
        Map QV: 50
  Query strand: 0
 Target strand: 1
   QueryRange: 278 -> 295 of 598
  TargetRange: 4469640 -> 4469657 of 5278900
    278  AGAGATAACCAGCAAGA
         |||||||||||||||||
4469640  AGAGATAACCAGCAAGA

aligning interval : 598 2659891 2659907 214 230 2659891 to 2659907 1.3
interval align score: -35
         Query: SRR305922.5 length=598/0_598
        Target: TY-2482_chromosome
         Model: a hybrid of global/local non-affine alignment
     Raw score: -35
        Map QV: 50
  Query strand: 0
 Target strand: 0
   QueryRange: 85 -> 92 of 598
  TargetRange: 2660363 -> 2660370 of 5278900
     85  CGCAGTA
         |||||||
2660363  CGCAGTA
```

FIGURE 4.3: A segment of the output of the BLASR aligner with the verbose option used.

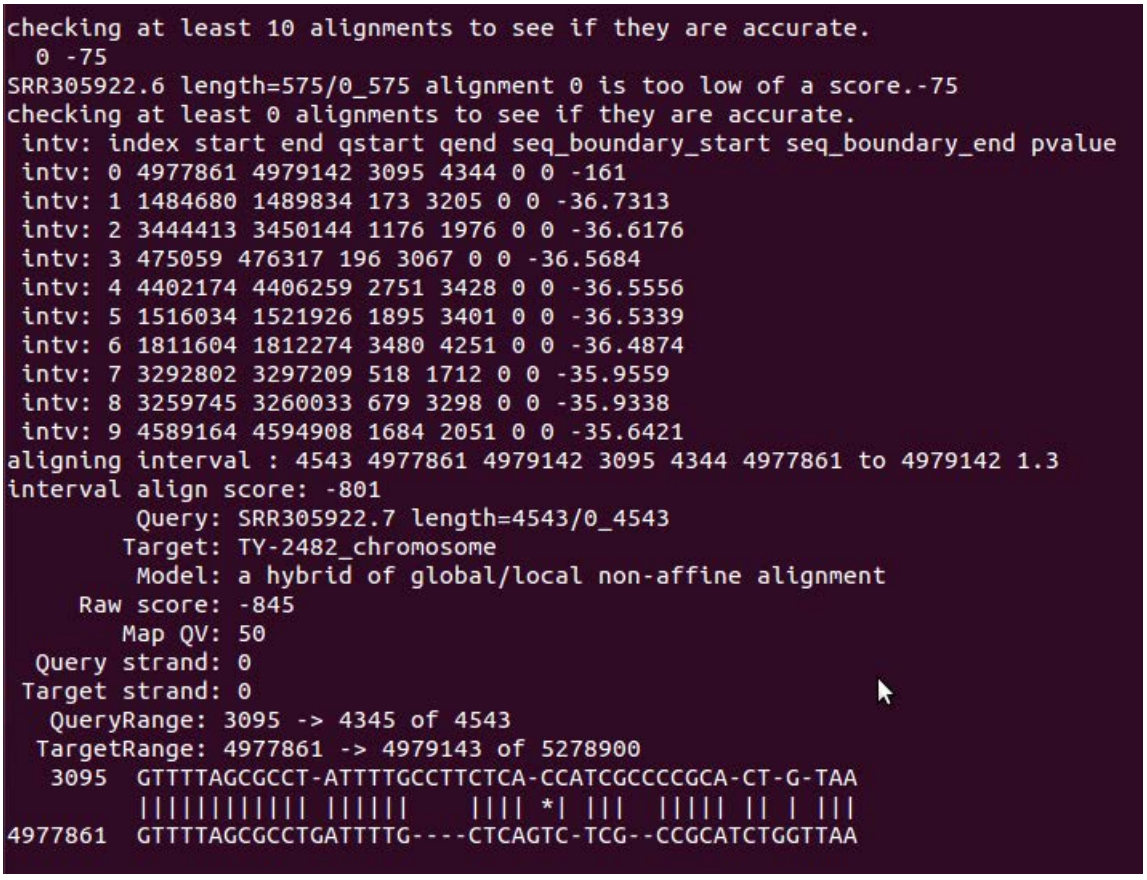

FIGURE 4.4: Another segment of the verbose output showing the reads that were aligned to the reference genome.

More details of the procedure to obtain the outputs are in *Appendix B*.

Chapter 5

Conclusions

In the almost three-month time spanned by the field experience, a lot of experience was gained, sometimes more than what was expected. Even though the activities performed went in accordance with the expected learning outcomes, a lot of lessons were learned indirectly while performing the activities. First, working individually and learning new things, as well as learning how to learn them, was practiced extensively. Learning topics and concepts that require background from another field (molecular biology in this case) was not a trivial task, especially when deciding how much information is necessary to learn or how deep the subject should be explored. Hence, the way of learning was developed into going to the breadth of the topic first, and then to the depth of its parts as necessary, while building assumptions throughout the process. Second, working as a team to perform a task, while coordinating with team members and the supervisor was also practiced well. This was beneficial since efficient ways were developed to be doing the job while working in parallel to document the work done. In addition, the communication skills were enhanced. This also applies when some of the activities practiced required direct communication with the professionals in the field to inquire about the subject material. The communication was not only a chance to be closely exposed to the professional environment, but also a chance to learn the way of asking questions to get the most out of the answers. Equally important are the lessons learned while experiencing the nature of doing research, such as working to create benchmarks.

No matter how challenging the internship was, the experiences that were gained, as well as the direct and indirect lessons learned, made it valuable.

References

[1] Andrews, S. "FASTQC. A quality control tool for high throughput sequence data." URL `http://www.bioinformatics.babraham.ac.uk/projects/fastqc` (2010).

[2] Chaisson, Mark J., and Glenn Tesler. "Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory." *BMC bioinformatics* 13, no. 1 (2012): 238.

[3] Kentie, Marijn. "Biological sequence alignment using graphics processing units." PhD diss., M. Sc. thesis, Computer Engineering Laboratory, Technical University Delft, The Netherlands, 2010.

Appendix A

Quality Control and Filtering of Reads

(Part of the Alignment Process)

In this appendix, a part of the alignment process, which is quality control and filtering, is shown. This part includes reviewing the raw sequence data of a yeast strain *Saccharomyces cerevisiae* and performing quality control to the reads. The programs and the genetic data were provided with the practical tutorial during the internship. The tutorial was performed on a Linux Knoppix system, which had all the data and programs already installed. The system can be downloaded as an ISO image from `http://helix.ewi.tudelft.nl/drop/knoppix_ac_final.iso`.

First, the mechanism of evaluating the quality of the reads by viewing them in FASTQ format will be shown. Then, filtering will be performed to the reads using *trimmomatic* program. Finally, the Fast QC program will be used to study the quality of the filtered reads.

A.1 Opening Read Files in FASTQ Format

To perform quality evaluation of the raw reads, the file containing the reads is opened in the terminal and viewed in FASTQ format. There is a certain way to interpret the data in FASTQ, which can be viewed from `http://en.wikipedia.org/wiki/FASTQ_format`.

The directory in which the raw illumina sequence files for the yeast strain CEN.PK113-7D (cenpk-chr3) and the evolved strain imw004 in FASTQ format (.fastq) is opened from the terminal, as shown in Figure A.1.

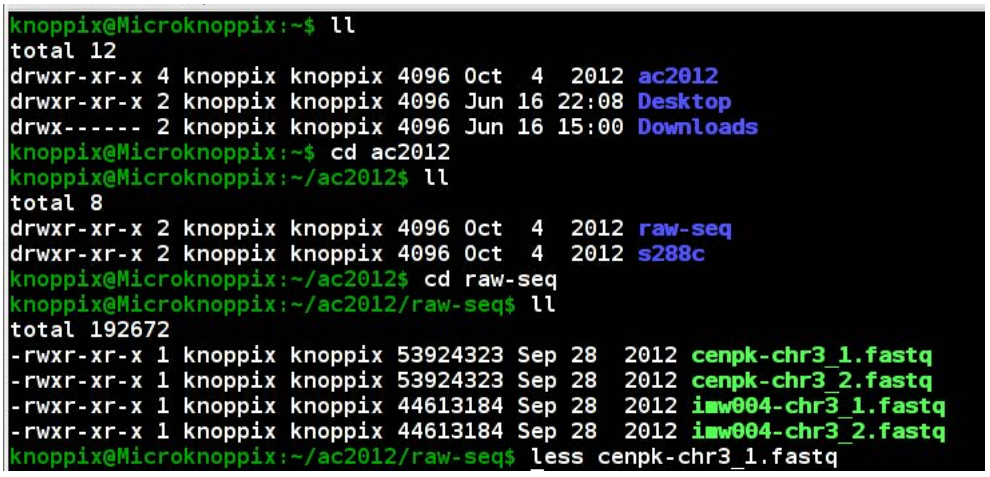

FIGURE A.1: Accessing the CEN.PK113-7D sequence

The last command in Figure A.1 is entered in order to read the raw reads file and view them in .fastq format. When the command is executed, the output looks like the one shown in Figure A.2.

FIGURE A.2: The raw CEN.PK113-7D sequence file in .fastq format

The file in FASTQ format lists information about the reads. Each read is composed of four lines. The first line is the sequence identifier that contains the name specific to that read. The second line contains the raw sequence letters, which are the bases (nucleotides) that compose the read. The third line contains a '+' character, which can optionally be followed by the the same sequence identifier, or any description. The fourth line contains encoded characters that represent the quality values of the sequence in the second line (one character corresponds to one nucleotide). Figure A.3 contains an annotation of each line that appears in the output.

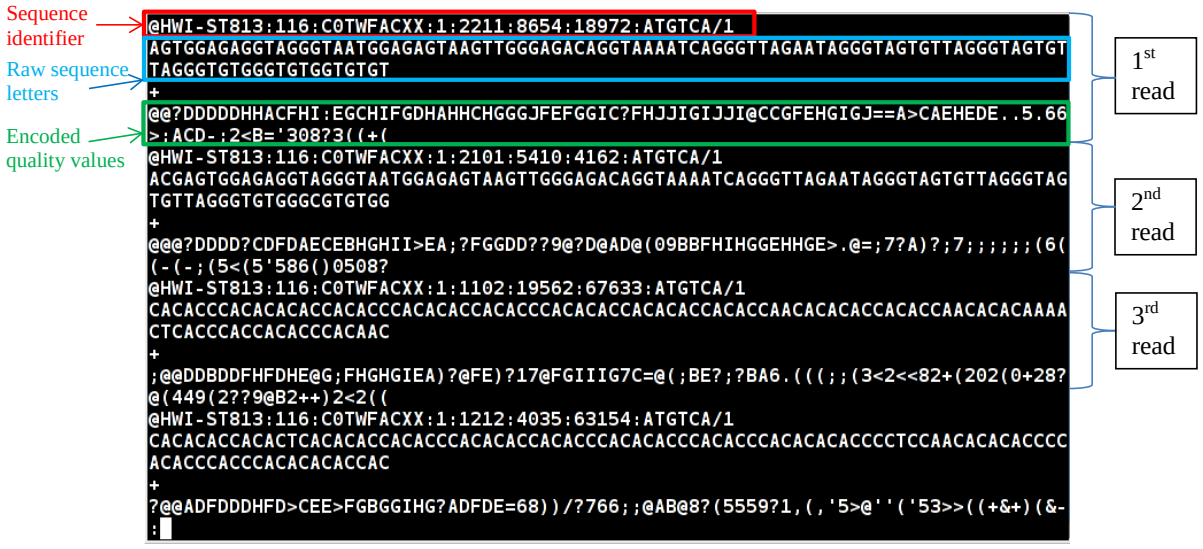

FIGURE A.3: The .fastq file annotated

The quality score of each base can be evaluated by knowing what the corresponding character in the fourth line encodes to in ASCII. [CITATION NEEDED FROM THE WIKIPEDIA SOURCE OF HOW TO READ FASTQ FILES]. For example, if the quality score of the fourth base in the second read is to be calculated, the procedure would be as follows:

This is the second read in FASTQ format:

```
@HWI-ST813:116:C0TWFACXX:1:2101:5410:4162:ATGTCA/1
ACGAGTGGAGAGGTAGGGTAATGGAGAGTAAGTTGGGAGACAGGTAAAATCAGGGTTAG
AATAGGGTAGTGTTAGGGTAGTGTTAGGGTGTGGGCGTGTGG
+
@@@?DDDD?CDFDAECEBHGHII>EA;?FGGDD??9@?D@AD@(09BBFHIHGGEHHGE
>.@=;7?A)?;7;;;;;;(6((-(-;(5<(5'586()0508?
```

The fourth base in the second read, which is A, has a corresponding quality symbol '?' (shown in bold above). This symbol, which is a character in ASCII format, is equivalent to 63 in decimal. Illumina 1.8 or higher encodes using Sanger format, which can encode a quality score of a range 0 to 93 using ASCII 33 to 126. With a decimal value of 63, the base has a Phred quality score of $63 - 33$, which is 30. So $Q = 30$ and since $Q = -10 \log_{10} P$,

$P = 10^{-Q/10} = 10^{-30/10} = 1/1000$. P, which is the probability of incorrect base call, is 1 in 1000.

A.2 Quality Evaluation using FastQC

Modern high throughput sequencers can generate tens of millions of sequences in a single run. Before analyzing this sequence to draw biological conclusions, quality control should be performed to ensure that the raw data has no problems or biases [1]. Hence, FastQC software is used to inspect the reads and produce QC reports. In those QC reports, problems originating from either the sequencer or in the starting material can be detected. The FastQC program can be downloaded from `http://www.bioinformatics.babraham.ac.uk/projects/fastqc/`. The quality results for the reads of the yeast strain sequence obtained from the FastQC program are shown in Figures A.4 to A.12.

First, the Basic Statistics module generates some simple composition statistics for the file analyzed:

- Filename: The original filename of the file which was analysed
- File type: Says whether the file appeared to contain actual base calls or colorspace data which had to be converted to base calls
- Encoding: Says which ASCII encoding of quality values was found in this file.
- Total Sequences: A count of the total number of sequences processed. There are two values reported, actual and estimated. At the moment these will always be the same.
- Filtered Sequences: If running in Casava mode sequences flagged to be filtered will be removed from all analyses. The number of such sequences removed will be reported

here. The total sequences count above will not include these filtered sequences and will the number of sequences actually used for the rest of the analysis.

- Sequence Length: Provides the length of the shortest and longest sequence in the set. If all sequences are the same length only one value is reported.
- %GC: The overall %GC of all bases in all sequences

The basic stats of the *saccharomyces cerevisiae* sequence is shown in Figure A.4.

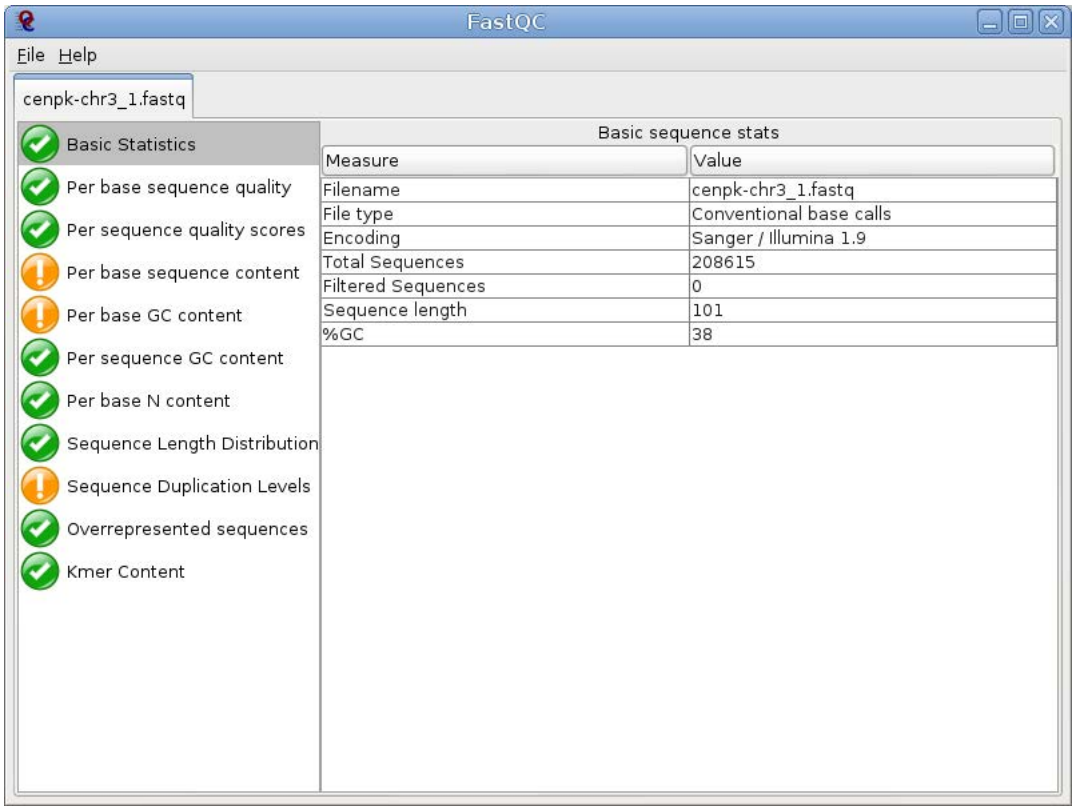

FIGURE A.4: The basic stats of the raw CEN.PK113-7D sequence. Notice how all the sequences have the same length (101 bases). None of those sequences is filtered.

Next is an overview of the range of quality values across all bases at each position in the FASTQ file, which is shown in Figure A.5.

For each position a BoxWhisker type plot is drawn. The elements of the plot are as follows:

- The central red line is the median value
- The yellow box represents the inter-quartile range (25-75%)
- The upper and lower whiskers represent the 10% and 90% points
- The blue line represents the mean quality
- The y-axis on the graph shows the quality scores. The higher the score the better the base call. The background of the graph divides the y axis into very good quality calls (green), calls of reasonable quality (orange), and calls of poor quality (red). The quality of calls on most platforms will degrade as the run progresses, so it is common to see base calls falling into the orange area towards the end of a read.[1].

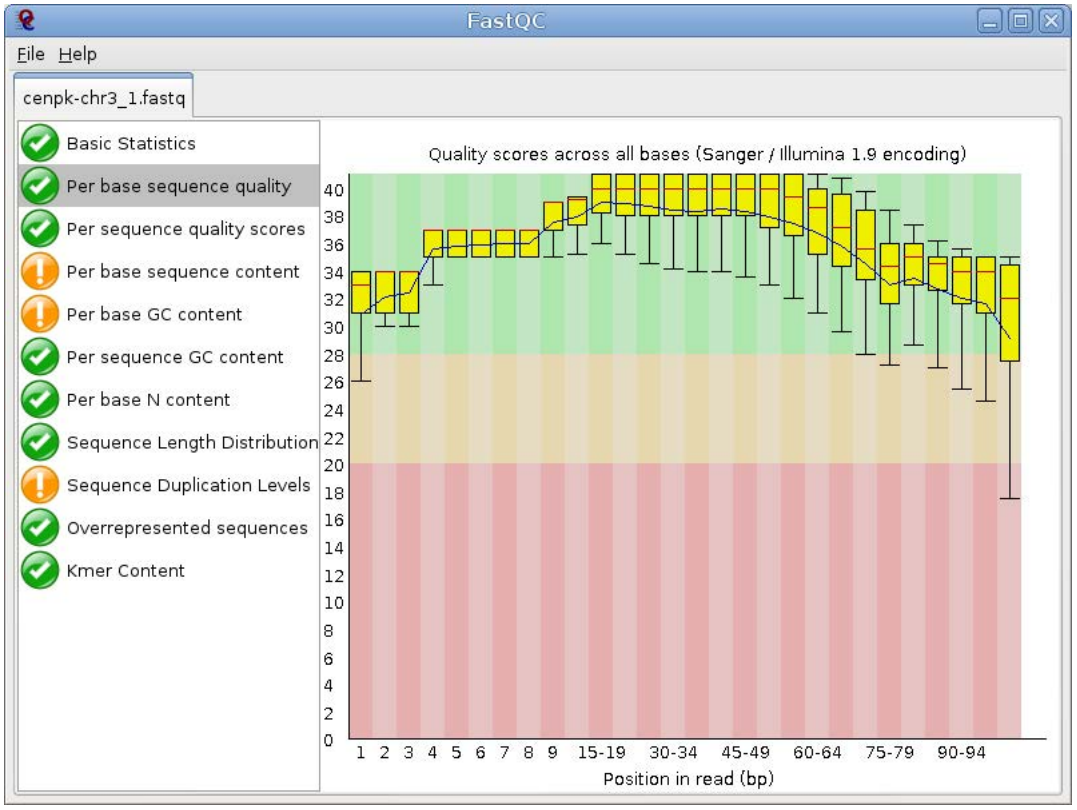

FIGURE A.5: The per base sequence quality of the raw CEN.PK113-7D sequence. Most of the bases have a high scores, signifying very good quality calls.

The per sequence quality score report (Figure A.6) checks for a subset of sequences having universally low quality values, which is happens often because they are poorly imaged (due to being on the edge of the field of view). However these should represent only a small percentage of the total sequences, otherwise warning or errors would show.

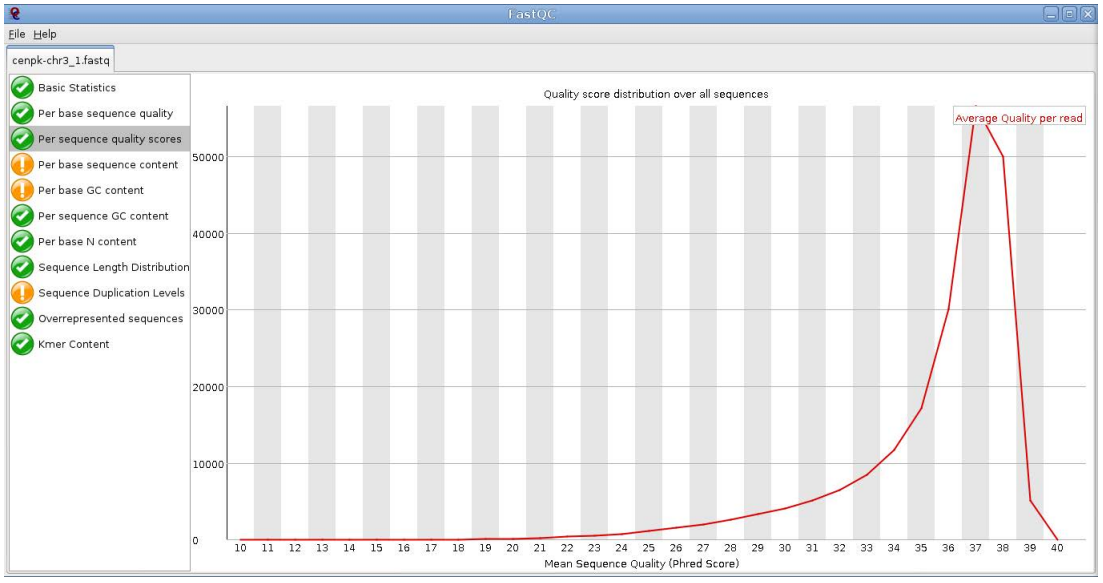

FIGURE A.6: The per sequence quality scores of the raw CEN.PK113-7D sequence. The general quality of this sequence is high.

Next, the per base sequence content plots out the proportion of each base position in a file for which each of the four normal DNA bases has been called.

There is not much difference between the different bases of a sequence (indicated by the parallel lines in this plot). The relative amount of each base reflects the overall amount of these bases in the genome, but in any case they should not be hugely imbalanced from each other (or sequence biased).

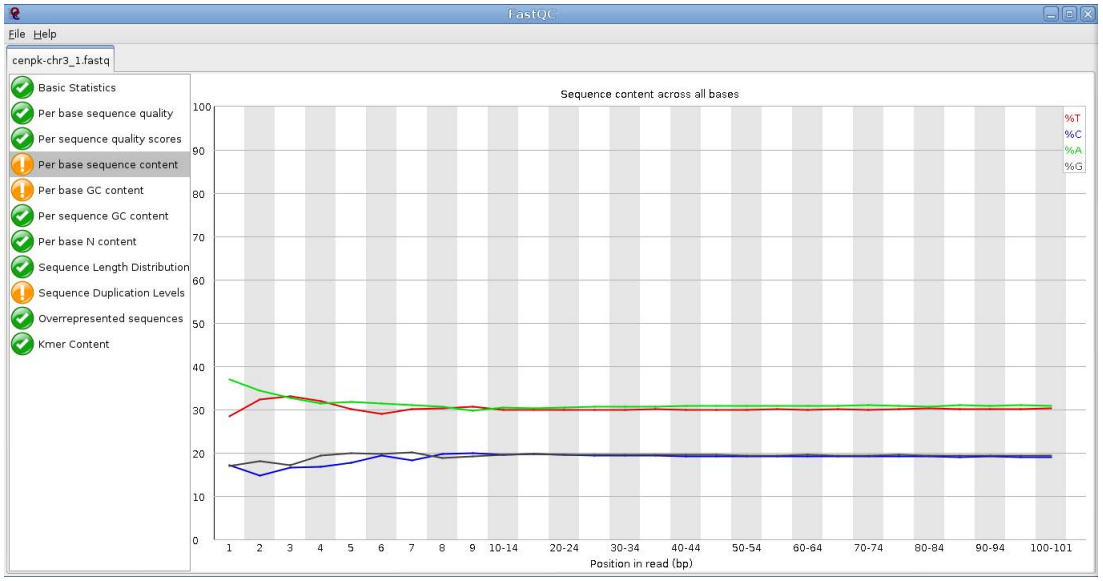

FIGURE A.7: The per base sequence content of the raw CEN.PK113-7D sequence. Notice how a warning is shown (by the orange sign near "per-base sequence content") because at some positions, the difference between the bases is greater than 10%.

The next graph is the per base GC content (Figure A.8), which plots the GC content of each base position in a file. Like any normal library, the sequence has very little difference between the different bases, which is shown by the horizontal line across the graph. If a GC bias is seen (when the GC content changes in different bases) then this could indicate an overrepresented sequence which is contaminating the library. A bias which is consistent across all bases either indicates that the original library was sequence biased, or that there was a systematic problem during the sequencing of the library.

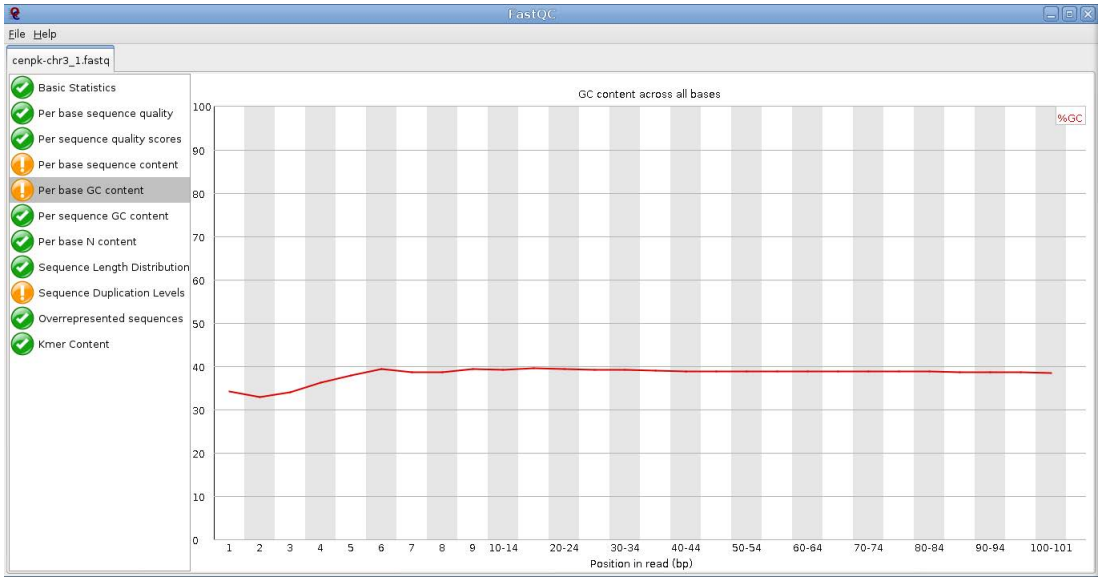

FIGURE A.8: The per base GC content of the raw CEN.PK113-7D sequence. Notice how a warning is shown module issues a warning because the GC content at the second base strays more than 5% from the mean GC content (which is 40%). An error would have shown if any of the bases strays more than 10% from the mean.

After the per base GC content, the GC content across the whole length of each sequence in a file is measured and compared to a modeled normal distribution of GC content. The GC content of this yeast strain roughly has a normal distribution, as shown in Figure A.9. The central peak corresponds to the overall GC content of the underlying genome. Since the GC content of the genome is not known the modal GC content is calculated from the observed data and used to build a reference distribution.

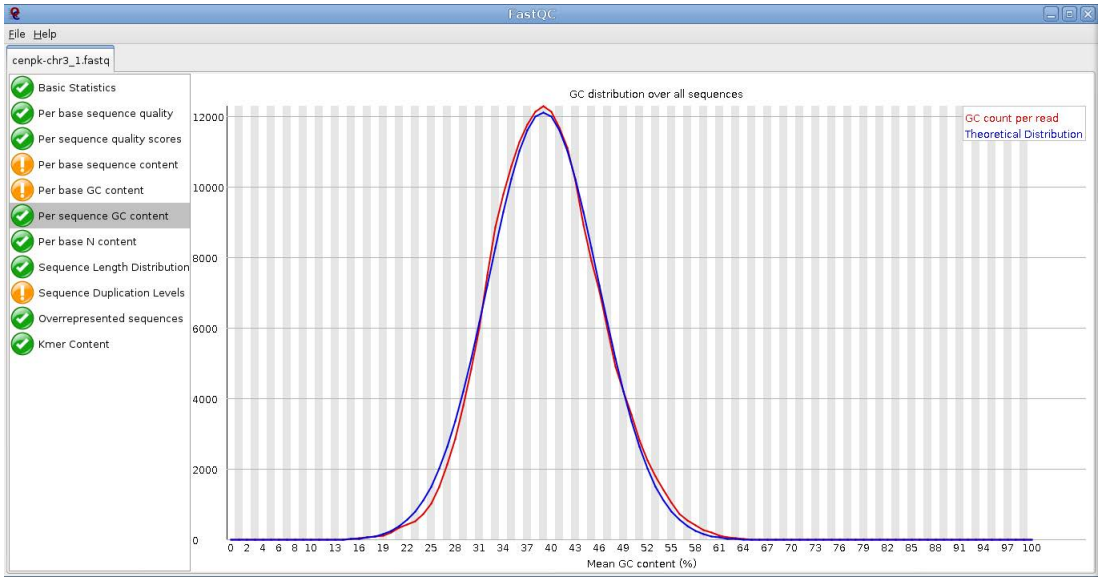

FIGURE A.9: The per sequence GC content of the sequence

Next is Figure A.10, which shows the per base N content. If a sequencer is unable to make a base call with sufficient confidence, then it will normally substitute an N rather than a conventional base call. This module plots out the percentage of base calls at each position for which an N was called. Notice the very low proportion of Ns appearing in the sequence (which is very usual). Yet, if this proportion rises above a few percent, it suggests that the analysis pipeline was unable to interpret the data well enough to make valid base calls.

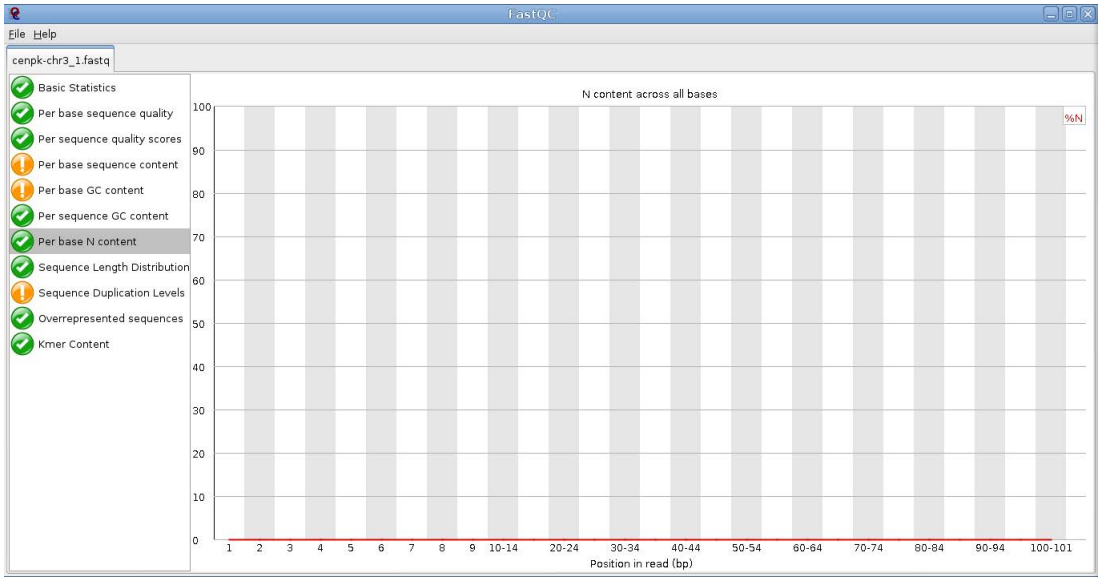

FIGURE A.10: The per base n content of the raw CEN.PK113-7D sequence

Some high throughput sequencers generate sequence fragments of uniform length, but others can contain reads of wildly varying lengths. Even within uniform length libraries, some pipelines will trim sequences to remove poor quality base calls from the end [1]. Figure A.11 shows the distribution of fragment sizes in the file being analyzed.

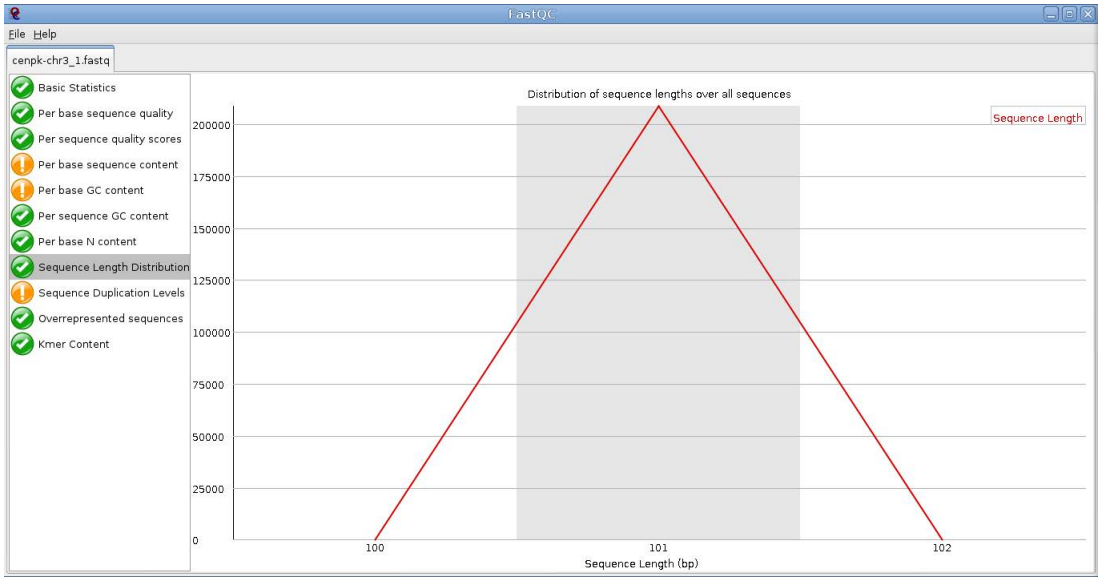

FIGURE A.11: The sequence length distribution of the raw CEN.PK113-7D sequence. Notice the peak only at the size of 101 because this is the length of each read.

In a diverse library most sequences will occur only once in the final set. A low level of duplication may indicate a very high level of coverage of the target sequence, but a high level of duplication is more likely to indicate some kind of enrichment bias Figure A.12 shows a plot showing the relative number of sequences with different degrees of duplication counted for every sequence in the set.

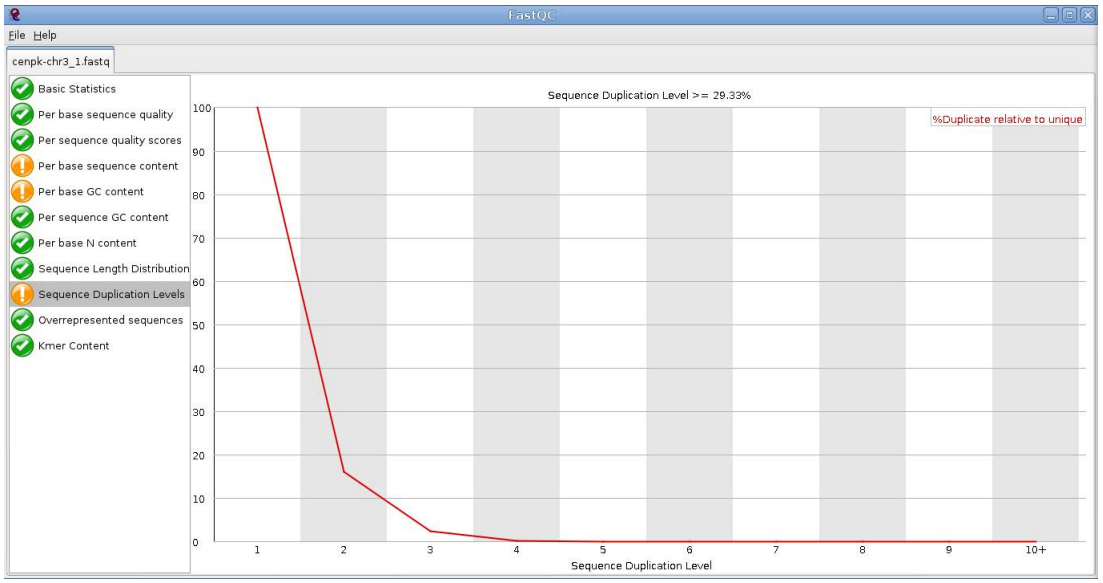

FIGURE A.12: The sequence duplication level of the raw CEN.PK113-7D sequence. A warning is issued because non-unique sequences make up more than 20% of the total.

A.3 Filtering using Trimmomatic

After performing quality analysis using the FastQC program, both cenpk-chr3_1 and cenpk-chr3_2 raw sequence reads were filtered with the following command:

```
knoppix@Microknoppix:~/ac2012/raw-seq$ trimmomatic.sh cenpk-chr3_1.fastq cenpk-c
hr3_2.fastq
TrimmomaticPE: Started with arguments: -phred33 cenpk-chr3_1.fastq cenpk-chr3_2.
fastq cenpk-chr3_1.fastq.trimmed_paired.fastq cenpk-chr3_1.fastq.trimmed_unpaire
d.fastq cenpk-chr3_2.fastq.trimmed_paired.fastq cenpk-chr3_2.fastq.trimmed_unpai
red.fastq LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:95
Input Read Pairs: 208615 Both Surviving: 176755 (84.73%) Forward Only Surviving:
 15897 (7.62%) Reverse Only Surviving: 8028 (3.85%) Dropped: 7935 (3.80%)
TrimmomaticPE: Completed successfully
knoppix@Microknoppix:~/ac2012/raw-seq$ ll
total 287948
-rwxr-xr-x 1 knoppix knoppix 53924323 Sep 28  2012 cenpk-chr3_1.fastq
-rw-r--r-- 1 knoppix knoppix 45649534 Jun 17 11:35 cenpk-chr3_1.fastq.trimmed_pa
ired.fastq
-rw-r--r-- 1 knoppix knoppix  4095843 Jun 17 11:35 cenpk-chr3_1.fastq.trimmed_un
paired.fastq
-rwxr-xr-x 1 knoppix knoppix 53924323 Sep 28  2012 cenpk-chr3_2.fastq
-rw-r--r-- 1 knoppix knoppix 45635934 Jun 17 11:35 cenpk-chr3_2.fastq.trimmed_pa
ired.fastq
-rw-r--r-- 1 knoppix knoppix  2068692 Jun 17 11:35 cenpk-chr3_2.fastq.trimmed_un
paired.fastq
-rwxr-xr-x 1 knoppix knoppix 44613184 Sep 28  2012 imw004-chr3_1.fastq
-rwxr-xr-x 1 knoppix knoppix 44613184 Sep 28  2012 imw004-chr3_2.fastq
knoppix@Microknoppix:~/ac2012/raw-seq$
```

FIGURE A.13: Filtering the CEN.PK113-7D sequence using trimmomatic

Using the numbers given in the output shown in Figure A.13, the percentage of the reads passing the filter can be calculated, as shown in Table A.1:

TABLE A.1: Percentage of reads passing the filter after trimming

Both forward and reverse reads passing	84.73%
Forward only passing	7.62%
Reverse only passing	3.85%
Total passing	96.20%

The total can be verified by checking the number of dropped input read pairs from Figure A.13, which is 3.80%.

FIGURE A.14: The trimmed sequence

The coverage of the trimmed sequence data with respect to the size of chromosome 3 (316kbp) can be calculated.

By looking at the genome statistics from the FastQC program (shown in Figure A.15), the coverage can be calculated using the De novo assembly coverage equation (Eq. A.1).

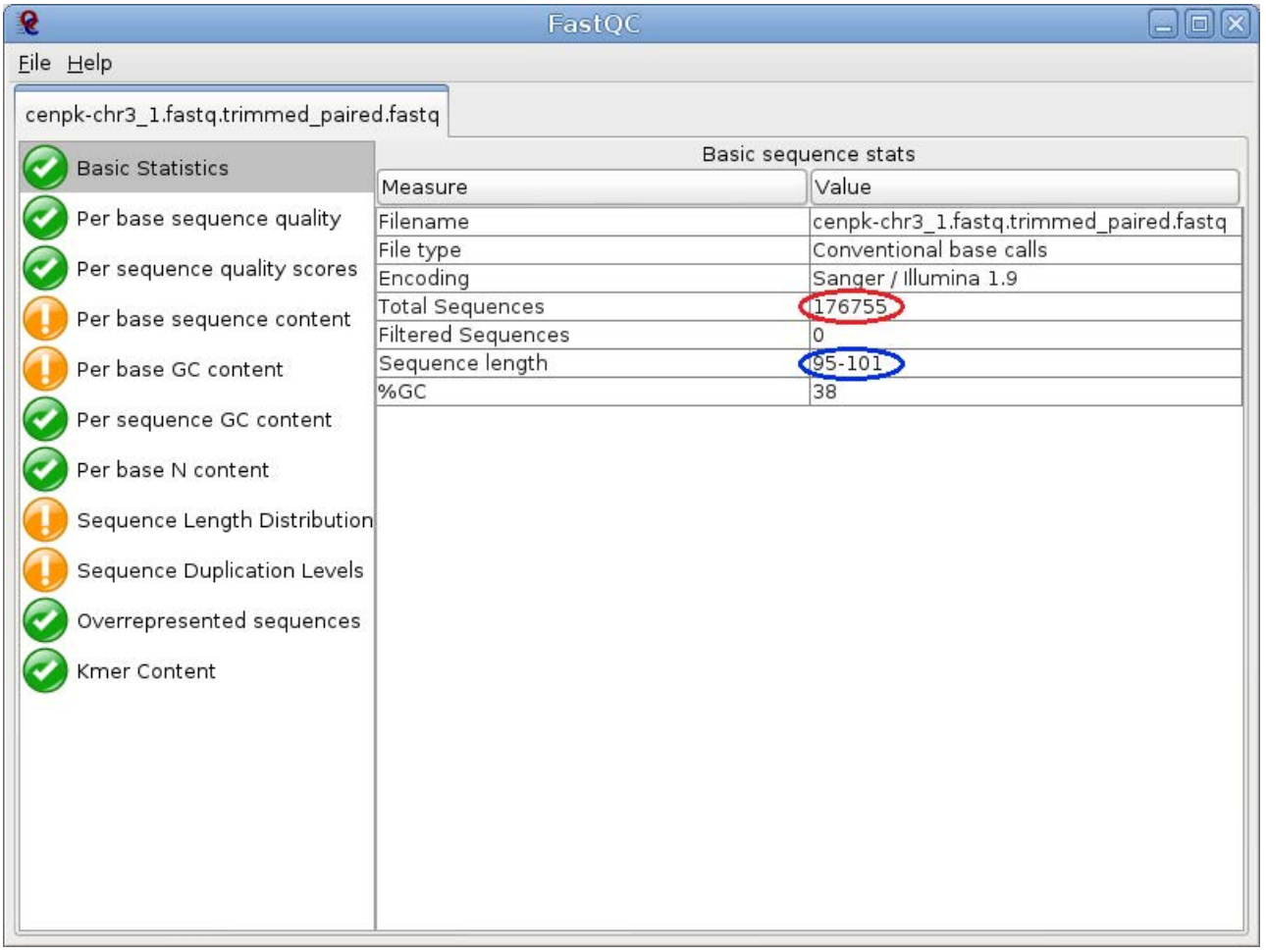

FIGURE A.15: The basic stats of the filtered CEN.PK113-7D sequence (after trimming)

$$\begin{aligned}\text{De novo assembly coverage} &= \frac{\text{number of reads} \times \text{read length} \times \text{number of files}}{\text{genome size}} \\ &= \frac{176755 \times 100 \times 2}{316000} \\ &\approx 111\times\end{aligned} \tag{A.1}$$

FIGURE A.16: The per base sequence quality of the trimmed CEN.PK113-7D sequence

Appendix B

Running BLASR Programs

This appendix contains guidance on how to download and run BLASR programs. It contains some screenshots of the commands and output.

B.1 Downloading BLASR

All the files comprising the BLASR program can be downloaded as a compressed folder from: `https://github.com/pacificbiosciences/blasr`.

Before running the programs, hdf 1.8.0 (or higher), which contains the library libhdf5_cpp.a should be installed. This can be done by running the following command from the linux terminal:

```
sudo apt-get install libhdf5-serial-dev
```

The raw reads mentioned in the BLASR paper are the SRR30592* series of the E. coli. The SRR305922 sequence is downloaded from

`http://sra.dnanexus.com/runs/SRR305922`.

The other sequences are downloaded from the same website. The files that are downloaded are in .sra format. To be able to open them, they need to be converted to .fastq format, which is done by using the SRA toolkit, downloaded from `http://www.ncbi.nlm.nih.gov/Traces/sra/?view=software` (A 64-bit architecture operating system is needed). After extracting the folder, the bin file that appears is opened from the linux terminal. Then, the following command is ran:

```
./fastq-dump ../../SRR305922.sra
```

The data are now ready to be used in fastq format. The reference genome for these data sets is the Ecoli_TY-2482 and can be downloaded from the following link: `ftp://ftp.genomics.org.cn/pub/Ecoli_TY-2482/Escherichia_coli_TY-2482.chromosome.20110616.fa.gz`

Before running BLASR, one compilation step is needed. Before compiling, it is necessary to point two environment variables, HDF5INCLUDEDIR and HDF5LIBDIR to the locations of the HDF5 libraries.

For example, type the following in the command line:

```
>export HDF5INCLUDEDIR=/usr/include/hdf
>export HDF5LIBDIR=/usr/lib/hdf
```

It is also necessary for g++ to be installed before compilation. After these steps are done, compilation can be completed by typing "`make`" in the command line while in the blasr-master directory. A "bin" folder will then be created in "/blasr-master/alignment/". BLASR can be run from the /blasr-master/alignment/bin directory. Typing the following will result in a help screen with a list of all options regarding the usage of BLASR:

```
./blasr -h
```

A help menu will appear as shown in Figure B.1.

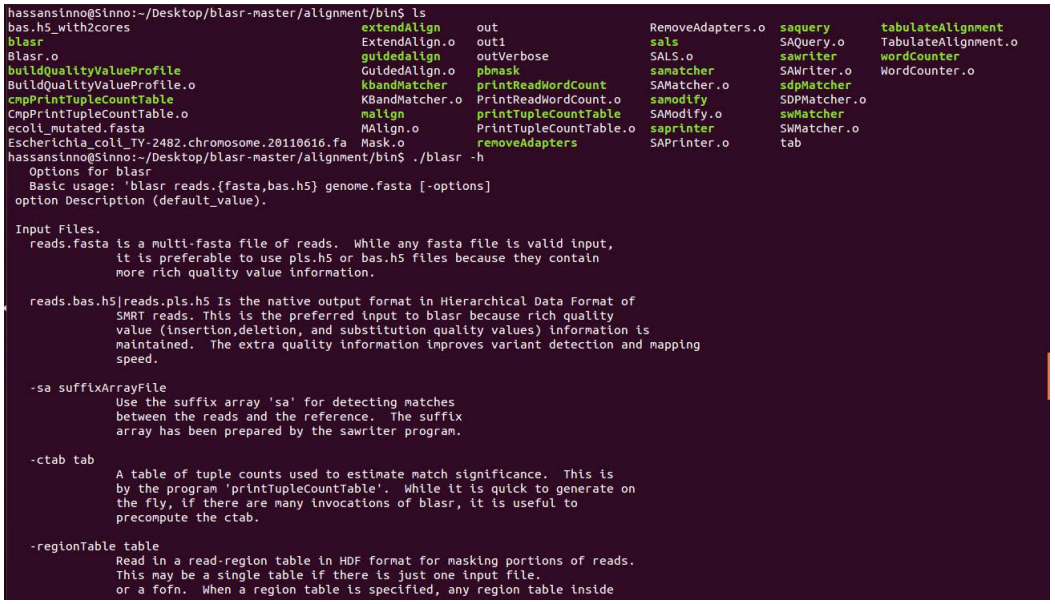

FIGURE B.1: BLASR folder and help menu

B.2 Running BLASR

A good idea would be to move the read files and reference genomes to the BLASR directory as this will simplify the commands to be typed in the Linux terminal. An example command would be:

```
mv ../../../technote-ecoli-native-raw-reads-1.3.1/2530179/0001\
/Analysis_Results/m120510_023815_42142_c100315192550000000152301\
3408241240_s1_p0.fastq .
```

This assumes that the downloaded reads folder is in the same directory as the "blasr-master" folder.

BLASR needs at least a read file and a reference genome to start aligning. Some options can be added to make the output more readable. Using `-header` will print a first row with the description of each column of data. Using `-titleTable table` will create a text file "table" containing the name of the reference genome; this will remove its name

from the output file, which makes reading it much easier (especially if the name of the reference genome is large). Also, using `-out outputFile` will print the output inside a text file "outputFile" instead of displaying the results on the Linux terminal. Finally, using `-nproc 4` will make use of four processors on your computer, which will reduce the run time significantly.

Thus, typing

```
./blasr m120510_023815_42142_c100315192550000001523013408241240\
_s1_p0.fastq ecoli_mutated.fasta -header -titleTable table -out\
out
```

would result in the output show in Figure B.2.

```
qname tname qstrand tstrand score pctsimilarity tstart tend tlength qstart qend qlength ncells
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/7/0_1326 0 0 1 -784 76.5101 1567130 1567374 4639560 445 733 1326 6421
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/9/0_2570 0 0 0 -2119 91.9132 4597406 4597900 4639560 2010 2495 2570 10291
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/9/0_2570 0 0 1 -304 96.9231 42083 42148 4639560 1895 1959 2570 1316
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/15/0_4383 0 0 1 -1723 89.3182 231109 231531 4639560 976 1394 4383 8893
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/15/0_4383 0 0 0 -1444 85.8911 4408038 4408416 4639560 488 867 4383 8149
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/16/0_4697 0 0 0 -1751 89.1111 2233241 2233677 4639560 951 1375 4697 9016
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/16/0_4697 0 0 1 -1461 91.9771 2405922 2406262 4639560 518 852 4697 7063
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/25/0_5541 0 0 1 -2863 93.3535 2447906 2448550 4639560 1969 2612 5541 13602
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/25/0_5541 0 0 0 -1779 91.784 2191229 2191630 4639560 4196 4613 5541 8892
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/46/0_4799 0 0 1 -2785 93.8679 491367 491983 4639560 26 648 4799 13159
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/46/0_4799 0 0 0 -2645 91.6667 4147581 4148195 4639560 697 1307 4799 12967
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/48/0_4195 0 0 1 -1927 85.7671 4615100 4615627 4639560 3000 3486 4195 10413
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/48/0_4195 0 0 0 -1350 82.8502 23826 24230 4639560 3818 4181 4195 7812
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/50/0_5032 0 0 1 -667 81.9048 4529113 4529316 4639560 4175 4357 5032 3908
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/50/0_5032 0 0 0 -393 88.3495 110244 110344 4639560 2763 2859 5032 2019
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/59/0_3979 0 0 0 -203 78.3784 2996165 2996235 4639560 1149 1218 3979 1460
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/62/0_4930 0 0 1 -2362 92.4596 3889367 3889902 4639560 1108 1648 4930 11460
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/62/0_4930 0 0 0 -2211 90.2174 749655 750191 4639560 506 1029 4930 11120
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/77/0_370 0 0 0 -414 88.785 2896041 2896142 4639560 4 106 370 2147
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/64/0_5654 0 0 1 -1075 85.2941 3637265 3637565 4639560 1211 1483 5654 5824
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/64/0_5654 0 0 0 -617 87.8049 1002184 1002345 4639560 3269 3419 5654 3174
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/93/0_744 0 0 0 -1571 89.0819 2024219 2024602 4639560 239 622 744 8154
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/93/0_744 0 0 1 -751 85.1163 2615136 2615341 4639560 0 197 744 4204
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/94/0_8254 0 0 1 -1535 77.6386 4381097 4381565 4639560 6942 7477 8254 11787
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/98/0_8737 0 0 0 -271 87.6712 3116203 3116274 4639560 2489 2559 8737 1453
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/99/0_7215 0 0 0 -640 90.566 3919763 3919916 4639560 5144 5299 7215 3265
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/100/0_6273 0 0 0 -2131 90.6667 3457572 3458082 4639560 3329 3824 6273 10535
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/100/0_6273 0 0 1 -1687 89.8585 1181557 1181969 4639560 2868 3264 6273 8424
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/104/0_3412 0 0 1 -1088 88.6525 396472 396748 4639560 2266 2524 3412 5480
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/104/0_3412 0 0 0 -989 89.9598 4242853 4243096 4639560 547 783 3412 4989
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/107/0_253 0 0 1 -533 89.6296 1665159 1665287 4639560 17 147 253 2740
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/106/0_3865 0 0 1 -2587 79.0879 4085614 4086490 4639560 1287 2049 3865 18783
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/106/0_3865 0 0 0 -1789 80.8219 553029 553590 4639560 434 940 3865 10964
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/112/0_3713 0 0 1 -1266 89.6875 3665577 3665882 4639560 311 617 3713 6501
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/112/0_3713 0 0 0 -1235 85.6322 973657 973984 4639560 1430 1754 3713 6933
Plain Text ▾  Tab Width: 8 ▾  Ln 6, Col 108  INS
```

FIGURE B.2: BLASR output in .fasta format

Figure B.3 shows the output after running the command (using .fasta format). Notice how the `tname` column is has the actual reference genome name.

```
qname tname qstrand tstrand score pctsimilarity tstart tend tlength qstart qend qlength ncells
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/5/0_204 ref000001|ecoliK12_mutated 0 1 -751 89.1192 2832262 2832451 4639560 2
182 403 3847
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/9/0_842 ref000001|ecoliK12_mutated 0 0 -210 90.3846 4597563 4597610 4639560 149
201 2570 1086
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/9/893_1966 ref000001|ecoliK12_mutated 0 1 -304 96.9231 42083 42148 4639560 1895
1959 2570 1317
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/9/2010_2540 ref000001|ecoliK12_mutated 0 0 -2260 91.1232 4597406 4597944 4639560
2010 2537 2570 11191
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/4/48_795 ref000001|ecoliK12_mutated 0 1 -3016 88.7324 1219877 1220615 4639560 50
795 5049 16006
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/4/840_1568 ref000001|ecoliK12_mutated 0 0 -3002 89.2068 3418945 3419685 4639560
840 1568 5049 15540
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/4/1611_2339 ref000001|ecoliK12_mutated 0 1 -3215 92.7056 1219877 1220611 4639560
1611 2335 5049 15347
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/4/2378_3133 ref000001|ecoliK12_mutated 0 0 -3092 90.0517 3418946 3419679 4639560
2380 3126 5049 15902
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/4/3177_3896 ref000001|ecoliK12_mutated 0 1 -3021 90.2926 1219884 1220615 4639560
3187 3896 5049 15094
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/4/3940_4655 ref000001|ecoliK12_mutated 0 0 -3005 89.3979 3418946 3419683 4639560
3940 4655 5049 15257
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/4/4697_5048 ref000001|ecoliK12_mutated 0 1 -1469 90.411 1219884 1220240 4639560
4703 5048 5049 7314
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/15/0_428 ref000001|ecoliK12_mutated 0 1 -1645 86.0262 231111 231544 4639560 3
427 4383 9090
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/15/470_916 ref000001|ecoliK12_mutated 0 0 -1657 85.2008 4408015 4408458 4639560
470 911 4383 9500
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/15/960_1404 ref000001|ecoliK12_mutated 0 1 -1758 88.9625 231098 231531 4639560
964 1394 4383 9151
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/15/1447_1868 ref000001|ecoliK12_mutated 0 0 -1717 87.9121 4408015 4408462
4639560 1448 1864 4383 8871
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/15/1914_2332 ref000001|ecoliK12_mutated 0 1 -1614 84.7639 231096 231546 4639560
1915 2332 4383 8979
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/15/2375_2807 ref000001|ecoliK12_mutated 0 0 -1791 89.2779 4408017 4408459
4639560 2378 2805 4383 9092
m120510_023815_42142_c100315192550000001523013408241240_s1_p0/16/187_435 ref000001|ecoliK12_mutated 0 0 -969 86.194 2233424 2233677 4639560
```

FIGURE B.3: BLASR output in .bas.h5 format

More options can be tried from the help menu (shown in Figure B.1) to produce different outputs.

Appendix C

Learning Outcomes **(From Course Syllabus)**

Following is the list of the learning outcomes (1, 4, 5, 6, 7, and 8) that must be addressed by Chapter 3. *Field Experience Activities*, according to the course syllabus.

Upon completion of this course students will be able to:

1. Design computer engineering components and processes to satisfy technical and non-technical constraints, as specified by the field supervisor or client.

4. Function as a team member on multidisciplinary teams in a real workplace setting.

5. Cite specific examples of ethical and professional issues encountered during the field experience.

6. Outline the global, economic, environmental, or societal impact of the engineering activities carried out during the field experience.

7. Discuss examples of life-long learning and professional development activities in the workplace.

8. Describe contemporary and emerging issues pertinent to the industry in which the field experience was completed.