QUALITY ASSESSMENT OF DE NOVO SEQUENCE ASSEMBLY TOOLS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Molecular Biology and Genetics

by
Visam GÜLTEKİN

December 2012
İZMİR

We approve the thesis of **Visam GÜLTEKİN**

Examining Committee Members:

____

Assoc. Prof. Dr. Jens ALLMER
Department of Molecular Biology and Genetics,
İzmir Institute of Technology

____

Prof. Dr. Anne FRARY
Department of Molecular Biology and Genetics
İzmir Institute of Technology

____

Assoc. Prof. Dr. Bilge KARAÇALI
Department of Electrical - Electronics Engineering
İzmir Institute of Technology

12 December 2012

____

Supervisor, Assoc. Prof. Dr. Jens ALLMER
Department of Molecular Biology and Genetics,
İzmir Institute of Technology

____

Assoc. Prof. Dr. Ahmet KOÇ
Head of the Department of
Molecular Biology and Genetics

____

Prof. Dr. R. Tuğrul SENGER
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

I would like to acknowledge my supervisor, Assoc. Prof. Jens Allmer, for his understanding and tolerance, precise guidance and intense support from the very start of the thesis study to the end.

It was a great opportunity; that professors of The Laboratory of Plant Molecular Genetics, Prof. Dr. Anne Frary and Prof. Dr. Sami Doğanlar generously supplied the data that was needed for the thesis study. I would like to thank them and MSc. İbrahim Çelik from The Laboratory of Plant Molecular Genetics, not only for providing the data but also because they were always open for me.

It has been and will be an honor to be a member of jLab; IZTECH Bioinformatics Laboratory. I am greatly appreciative to all members of jLab, for their support and efforts, more importantly their delicate friendship.

Last but totally not the least, I would like to thank my beloved wife, and her ever ready smile, who always was encouraging and supporting me.

ABSTRACT

QUALITY ASSESSMENT OF DE NOVO SEQUENCE ASSEMBLY TOOLS

High-throughput next generation sequencing technologies progressed very rapidly; revolutionized genomics by providing a robust working field for new studies to be performed and promising the facilitation of the achievements that was extremely challenging before. Although the massive output of these instruments is getting more accurate, still delivers the projection of the real sequence in very short fragments; which necessitates another process of merging and ordering those fragments to reconstruct the larger sequences. This process is performed by sequence assemblers and in the absence of a reference genome; it becomes a de novo sequence assembly. Since assembling millions of fragments in biological aspects have many obvious challenges, there have been many studies specifically focused on developing tools that can adapt to newly announced sequencing technologies, take advantage of the computer science achievements and the technological advancement of computer hardware to the utmost. But these sequence assemblers also need to justify the gain they claim. We took 5 of the commonly used assemblers and assembled two genomic datasets, mined the never mentioned statistics before and commonly used statistics that thought to be the representative of the quality of the assembly. On top of that we also used experimentally validated data that is known to be a part of the organisms' genome and trailed those in assemblies.

ÖZET

DE NOVO SEKANS MONTAJ ARAÇLARININ KALİTE DEĞERLENDİRMESİ

Yeni nesil sekanslama teknolojileri inanılmaz bir hızla gelişti; yeni çalışma alanları için gerekli zemini hazırlayarak özellikle genomik biliminde çığır açtı ve önceleri fazlasıyla zor olarak görülen birçok gelişmenin gerçekleştirilmesine olanak tanıdı. Bu teknolojinin sunduğu muazzam veri sürekli daha da hassas ve hatasız olsa da, biyolojik diziyi halen çok küçük boyutlu parçalar halinde vermekte ve bu durum daha büyük dizilerin oluşturulması için küçük dizilerin sıralanıp birleştirilmesinden oluşan bir başka uygulamanın gerçekleştirilmesini gerekli kılmaktadır. Bu uygulama dizi montajlama araçları tarafından yerine getirilmektedir ve referans genomun eksikliği durumunda de novo sekans montajlama işlemi haline gelir. Milyonlarca küçük boyutlu diziden biyolojik kriterlere uygun şekilde daha büyük dizilerin oluşturulmasının önünde birçok zorluk yatmaktadır ve bu zorlukların aşılabilmesi için birçok bilimsel çalışma yapılmaktadır. Bilgisayar biliminin kazanımlarını, bilgisayar donanım teknolojilerinin eriştiği son noktayı kullanabilecek ve yeni duyurulmuş sekanslama teknolojileri ile uyumlu çalışacak araçlar tasarlanmaktadır. Bu araçların vadettiklerinin doğrulanması gereklidir. Bu bağlamda 5 adet yoğun kullanılan sekans montaj aracını incelemeye aldık; 2 bağımsız genomik veri ile dizi montajlaması yaptırdık. Dizi montaj kalitesinin dile getirilmesinde kullanılan genel istatiksel metriklerin yanı sıra daha önce bahsedilmemiş istatiksel yönden de inceledik. Bunun yanı sıra çalışmamızda kullandığım organizmanın genomunda var olduğu deneysel olarak ortaya konmuş verilerin elde ettiğimiz dizi montajlarında varlığı doğrulamak üzere çalışmalar gerçekleştirdik.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. Sequencing

Sequencing is the definition for a number of processes that try and determine the primary structure of a biological polymer. As a result, the sequencing process yields a **sequence -**that can be stored as ordinary text- and that represents the order of chemical building blocks of the sequenced molecule. This symbolic depiction of the molecule brings the possibility of virtual analysis, and brings the necessity of many scientific disciplines to analyze the information it presents (Hartl 1996; P a Pevzner, Tang, and Waterman 2001; a L. Delcher et al. 1999; Schuster 2008a; Enome and Equencing 1999). The availability of sequencing revolutionized biological studies especially concerning genes and genomes, and opened many more point of visions to the researchers by offering greater understanding of biology (a L. Delcher et al. 1999; W. MIN JOU, G. HAEGEMAN 1972, Ewing and Green 1998; Jaffe et al. 2003a).

Given the complexity and heterogeneity of biopolymers, sequencing techniques differ within biological molecules. Since sequencing techniques is beyond this study, we will just mention DNA, RNA (commonly described as transcriptome) and protein sequencing briefly.

1.1.1. Protein Sequencing

The process of determination of amino acid sequence in a protein molecule is called protein sequencing. Whilst amino acid sequence is determined, non-peptide molecules are also identified (P. C. Ng 2003). There are a number of methods of performing protein sequencing; of which mass spectrometry is the commonly used technique (Bocker 2003). Some of the major techniques are; Edman degradation, Peptide mass fingerprinting and Mass spectrometry.

1.1.2. RNA- Transcriptome Sequencing

RNA sequencing is commonly referred to as **transcriptome** (the set of all RNA molecules) **sequencing,** since RNA is generated from DNA in the cell with a process called transcription. RNA carries information which is readily present in the DNA, but especially in eukaryotes mRNA molecules will not include information of noncoding regions, such as introns, and that is among many reasons why RNA sequencing is performed (Barbazuk et al. 2007; Sugarbaker et al. 2008; Wu et al.; Malde, Coward, and Jonassen 2005). A commonly used technique for RNA sequencing is to reverse transcribe the RNA molecule, then apply vastly and commonly used DNA sequencing methods (Pauline C Ng and Henikoff 2003).

Transcriptome sequencing is essential in some scenarios to better understand the genome, and is referred to as "the most definitive approach to the elucidation of transcripts" especially for organisms that show higher complexity, like mammals (Bastien Chevreux 2005). This explaining the RNA sequencing being one of the earliest sequencing techniques; with the publish of the complete genome of the Bacteriophage MS2 by Fiers et. al (W. MIN JOU, G. HAEGEMAN 1972; Fiers et al. 1976).

1.1.3. DNA Sequencing

DNA sequencing consists of a number of processes that aim to result in a sequence of nucleotide bases in a DNA molecule. Although the structure of DNA was known since 1953 (Watson and Crick 1953), it took researchers more than 20 years to first come up with the techniques to retrieve the nucleotide bases of a DNA strand (Weigel et al. 1973; Sanger, Nicklen, and Coulson 1977; Petrov et al. 1981; Z. Zhang et al. 2000; Couronne et al. 2003b; Bastien Chevreux et al. 2004; Bastien Chevreux 2005; Bentley 2006; Shendure and Ji 2008a; Wheeler et al. 2008; Pop 2009a; Rodrigue et al. 2010; Zerbino and Birney 2008a).

The very first advancements in DNA sequencing were announced almost at the same time period: Walter Gilbert and Allan Maxam's "DNA sequencing by chemical degradation" in 1973 (Bentley 2006; Shendure and Ji 2008b) which revealed the nucleotide sequence of the lac Operator (Maxam 1973) and Frederick Sanger's "DNA sequencing with chain-terminating inhibitors" in 1975 (Sanger and Coulson 1975) which yielded the nucleotide sequence of bacteriophage φ X174 DNA (Sanger F 1977).

Although these two aforementioned techniques are categorized as the basic methods (Istrail et al. 2012; B Chevreux; Baker 2012; Narzisi and Mishra 2011; Sanger F 1977), Sanger's method is still used more often (Bastien Chevreux 2005). With the development of technology, DNA sequencing techniques became much more advanced with the conjunction of computer science in time; reading longer fragments, parallelize the reading process, etc. Being able to produce thousands and even millions of sequences at one run, these high-throughput technologies are called **next-generation sequencing** technologies (Dear et al. 1998; X Huang and Madan 1999; Z. Zhang et al. 2000; Steven 2002a; Pop et al. 2004).

There are many next-generation sequencing technologies available today, since details of those are beyond the scope of this study only some of these techniques are given in Table1.

Table 1. A comparison of next generation sequencing technologies. (Source: (Narzisi and Mishra 2011; Dear et al. 1998; Zerbino and Birney 2008b; Z. Li et al. 2012; Steven 2002b)).

*Sanger sequencing method was included for comparison.

	Pacific Bio	**Ion Torrent**	**454**	**Illumina**	**SOLiD**	**Sanger***
Read length (max)	2900 bp	200 bp	700 bp	250 bp	50+50 bp	900 bp
Accuracy	99%	98%	99.9%	98%	99.9%	99.9%
Reads per run	35-75K	5 million	1.3 million	3 billion	1.4 billion	N/A
Time per run (minutes)	<120	<120	<630	<14.400	<20.200	<180
Cost per 1 million raw bases (in US$)	$2	$1	<$4	<$0.15	$0.13	$2400
Advantages	Longest read length. Fast.	Fast.	Long read size. Fast.	High sequence yield.	Low cost per base.	Long individual reads.
Disadvantages	Low yield	Homopolymer errors.	Expensive. Homopolymer errors.	Equipment can be very expensive.	Slower than other techs.	More expensive and impractical for larger projects.

DNA sequencing can be performed for various studies; including SNP discovery-genotyping, DNA profiling etc. (Tucker, Marra, and Friedman 2009). Since this study has been performed on genomic DNA sequencing data gained from one of next generation sequencing technologies, we will mentioning genome sequencing briefly.

1.1.3.1. Genome Sequencing

Genome sequencing is performed to cover the complete DNA sequence information of a given organism. This procedure may span different types of DNA for various organisms, like chromosomal, mitochondrial or chloroplast DNA. Any biological sample that contains full copy DNA can be taken into consideration in the wet lab preparation processes.

Whole genome sequencing with next generation sequencing technologies goes back to 1994, when Fleischmann et al. published the complete genome of Haemophilus influenza,1.8 Mb (mega bases) (Scheibye-Alsing et al. 2009, Fleischmann et al. 1995). This is accepted as a landmark in the history of sequencing and especially whole genome sequencing, because while it was the first and only complete genome sequence of a free-living organism, it encouraged many other projects that revealed many organisms' complete genomes in a short time period (Scheibye-Alsing et al. 2009; Baker 2012; Steven 2002a).

High-throughput next generation sequencing technologies are one of the hottest topics of genomics studies by being labor and cost effective, as each runcan produce over 100 times more sequence data compared to the most sophisticated basic methods (Pareek, Smoczynski, and Tretyn 2011). Having much more information at lower costs and in short time periods provided concrete ground for research and gave rise to genome size studies. As of October 2012, there are 176 eukaryotic and 2106 prokaryotic organism genomes that have been completely sequenced and published (KEGG2, GOLD). Figure 1 represents genomic sequencing studies' accumulation over time.

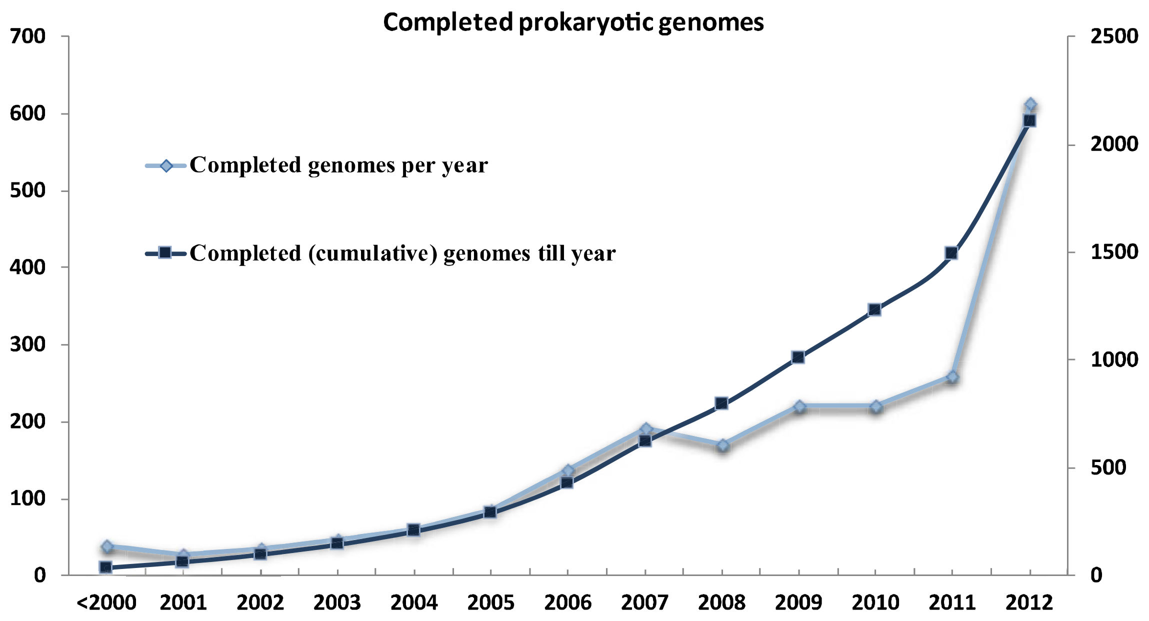

Figure 1. Completed prokaryotic genomes.

(Source: http://www.genome.jp/kegg/catalog/org_list.html, 2013.).

As already stated, recent improvements in sequencing technologies encourage researchers to perform studies that were dreams some time ago. One of the significant projects that deserves to be mentioned is the **1000 genomes (http://www.1000genomes.org) project** (Executive and Ceu 2007). Aiming to find genetic variants that would have at least 1% frequencies, the project covers the sequencing of a large number of people's DNA, in order to provide a comprehensive and valuable resource that will be accessible through public databases (Phase 2011).

1.2. Sequence Assembly

Sequence assembly refers to a number of processes including aligning and merging individual DNA fragments based on their sequence similarities gained from any sequencing technology into contiguous sequences called **contigs** for subsequent analysis (Sundquist et al. 2007; Walking 1998; Problem).

As seen in Table 1, each run of the DNA sequencing process makes available massive amounts of DNA sequence information, which need to be processed heavily and swiftly. This massive accumulation of data brought the inevitable need of automated work, performed by sophisticated bioinformatics tools that reduce the need of human involvement. These tools are at the point of conversion of the storehouses of information into knowledge and innovative solutions (Pareek, Smoczynski, and Tretyn 2011).

The amount of raw data generated from each run by next generation sequencing technologies is overwhelming. During the human genome project that was started in 1990 and finished in 2003, more than 50 billion bases of raw sequencing data were produced (International Human Genome Sequencing Consortium 23 Bb, Celera Projects 27 Bb) (Genomics 2004a; Venter et al. 2001). This massive raw data definitely demands high computational power that can result the analysis in reasonable time periods and more importantly accurately so that it can be further analyzed (Pavel a Pevzner et al. 2004; Couronne et al. 2003b).

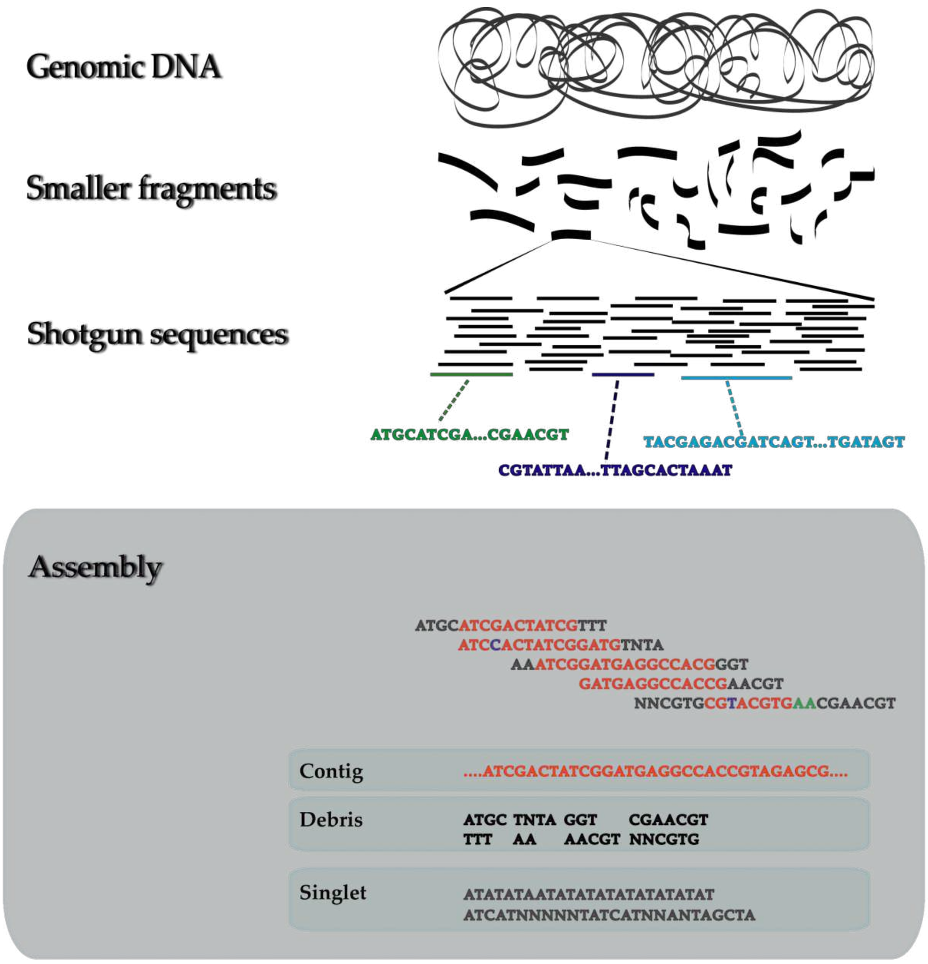

Figure 2. Shotgun sequences are aligned and merged with respect to their overlapping regions.

Figure 2 illustrates a quick procedure of real sequence assembly. For better understanding, it might be good to explain some definitons. A contig as mentioned before, is a sequence that has been generated by several to many smaller sequences. Singlets or singlet sequences are those which were not used in the step generating the contigs, hence left out of the assembly. Most of them are simply the sequences that keep the same before and after an assembly process. Debris sequences are leftovers from assembly processes, be it at the stage of sequence cleaning, ordering sequences or discorded parts of a sequence.

1.2.1. Sequence Assembly Tools

Sequence assembly studies generally require two major computational kinds of effort: assembling the data that has been generated by any kind of sequencing technologies and a finishing step that consist of some error correction processes, contig editing and further annotation (Palmer et al. 2010).

1.2.1.1. Assemblers

As the sequencing technologies developed very rapidly in time, new sequence assembly programs to provide, maintain and grasp the new criteria have been developed, existing ones got updated and revised (Lin et al. 2011b; Pop 2009a; Walking 1998; B Chevreux). One of the first assemblers was developed mainly to store and perform some limited manipulation, including clustering, for DNA gel fragments (Staden 1980). Starting from early 1990's, when larger eukaryotic genome studies started to become more attractive to researchers, several other assemblers were presented (Scheibye-Alsing et al. 2009). Several special efforts also have been announced, enriching assembler features and making assemblers more compatible with the reads (Scheibye-Alsing et al. 2009; Malde, Coward, and Jonassen 2005).

The very basic principle of assemblers' work is: overlapping fragments (which will be referred as reads from now on) might presumably originate from the same region of the original DNA sequence hence can be assembled together (Couronne et al. 2003a; X. Huang 1999; Narzisi and Mishra 2011; Steven L Salzberg and Yorke 2005; Steven 2002b).

This assumption is accepted globally yet brings the ambiguities of recognition of repeat sequences that might be present in many distinct places in the genome (Narzisi and Mishra 2011; Steven 2002a; Xiaoqiu Huang et al. 2003). One of the very important challenges that assemblers face is the error rate occurring from the nature of sequencing processes; for the good parts it would not exceed 2% while for bad parts it can quickly jump to around 10%. In order to tackle those and unmentioned other problems (like increased automation by reducing involvement in correcting errors), assemblers became more capable in time and moved away from simply using base sequences and onwards, using additional information like coverage analysis, original signal trace information, sequence orientation, template identity, quality and probability values (Bastien Chevreux et al. 2004).

Sequence assemblers can perform two different type of assembly: **de-novo** and **mapping**. De-novo assembly refers to the assembly type where there is no reference data of the organism that is studied; hence the assembly is performed with only read information. Mapping or reference-based assembly is aided by a previously known (by means of sequenced, studied earlier) backbone sequence, building contigs more or less identical to the reference (Kumar and Blaxter 2010a; Materials et al.; Z. Li et al. 2011; Chaisson, Pevzner, and Tang 2004).

Not all assemblers can perform these two types of assembly, but rather specialize their internal functions and capabilities to satisfy the needs. Also, assemblers specialize with respect to sequencing technologies; while some assemblers can perform accurate performance on just one of the sequencing technologies, some can adapt to work with several sequencing technologies with internal parameters (Tucker, Marra, and Friedman 2009; Ramos et al. 2011; Lin et al. 2011b; Pop 2009b; Narzisi and Mishra 2011; Istrail et al. 2012). Table 2 represents some of the assemblers with respect to their compatible technologies and type and size of assembly they can perform.

Table 2. Short list of sequence assemblers.(Source: (Narzisi and Mishra 2011; Shendure and Ji 2008a; W. Zhang et al. 2011)).

Name	Type- Size	Technologies
AMOS	genomes	Sanger, 454
Celera	(large) genomes	Sanger, 454, Solexa
CLC Genomics Workbench	genomes	Sanger, 454, Solexa, SOLiD
Euler-sr	genomes	454, Solexa
Geneious	genomes	Sanger, 454, Solexa
MIRA	genomes, EST	Sanger, 454, Solexa
NextGENe	small genomes	454, Solexa, SOLiD
Newbler	genomes, ESTs	454, Sanger
Phrap	genomes	Sanger, 454, Solexa
TIGR Assembler	genomic	Sanger
SeqMan NGen	genomes	Illumina, SOLiD, 454, Ion Torrent
SOAPdenovo	genomes	Solexa
Staden gap4	BACs	Sanger
Velvet	(small) genomes	Sanger, 454, Solexa, SOLiD

For several reasons like time and computational bottlenecks, this study covers only five of the sequence assemblers; -in alphabetical order- **CAP3** (X Huang and Madan 1999), **Celera** (Genomics 2004a), **MIRA** (Cancer 2005), **Newbler** (454 Life Sciences) (Margulies et al. 2005) and **Phrap** (Green 1997).

1.2.1.1.1. CAP3

CAP3 (http://pbil.univ-lyon1.fr/CAP3.php) is the third generation of the **CAP** sequence assembly program that was developed in 1992 (X Huang and Madan 1999). Adding the capability of identifying and clipping low quality 5’ and 3’ regions and improving the repeat recognition algorithm played a very good role in the program’s trending. Back before **CAP3** was announced, most assemblers were complaining after the difficulties of constraint usage in the assembling processes (X Huang and Madan 1999; Walking 1998). Being able to use forward-reverse constraints eased the process of assessment of DNA layout, hence results with consensus sequences that have fewer errors (X Huang and Madan 1999; Scheibye-Alsing et al. 2009; A. L. Delcher et al. 2002; Cancer 2005).

Like many assemblers **CAP3** has a preparation step that removes 5’ and 3’ poor quality regions. After this step comes the overlap computing. Contig construction is performed via a greedy fashion; a read with the highest scored overlap is taken and rest of the reads with less overlapping regions in a descending order. Then a multiple sequence alignment step takes place to build end contig sequences and after this step, consensus sequences with their base quality values are presented to user separately (X Huang and Madan 1999).

1.2.1.1.2. Celera

Celera is known to be the first assembler to report a complete human genome (Schatz 2006; Venter et al. 2012; Genomics 2004a) Of many assemblers that have been available, **Celera** is the one that has been developed by many scientists and took more than 20 years of effort to be completed (Schatz 2006).

Celera (http://wgs-assembler.sourceforge.net) as with most assemblers, starts the assembly processes by first identifying and tagging repeats (Genomics 2004b; Schatz 2006). Using signal traces and existing quality assessment of sequencing technology, **Celera** generates internal quality measures, especially to locate and trim the low quality regions that will be ignored till further steps (Genomics 2004a; A. L. Delcher et al. 2002). Being able to use additional information like trace signals, and aligning repeats to foreign contaminants like cloning-sequencing vectors and linkers, **Celera** aggressively removes, trims repeats, etc.(Myers et al. 2012).

After this step starts the overlap finding process, which is iterated in cycles to locate the longest overlaps possible, using a seed-and-extend algorithm like BLAST (Notredame, Higgins, and Heringa 2000; A. L. Delcher et al. 2002). At the first step of merging reads, pseudocontigs (**Celera** calls them unitigs) are compared to each other to find if unitigs represent longer repeats; if so those unitigs are also removed from calculations. The remainders are examined again, to find whether they contain any repetitive overlaps, detect and extend repeat boundaries (S. Li et al. 2002; Schatz 2006).

Having unitigs free of bad regions, starts the contig building step, and this is followed by a scaffolding step that is hoped to merge contigs and output longer end sequences (S. Li et al. 2002; Schatz 2006).

1.2.1.1.3. MIRA

MIRA’s development goes back to 1997, but the first publication was put forward in 1999 (Bastien Chevreux 2005). It is one of the assemblers that has been under development and adapted to newly announced sequencing technologies like IonTorrent in time. As of October 2012 **MIRA** 3.9.5. is distributed (http://MIRA-assembler.sourceforge.net).

Unlike many sequence assemblers available, **MIRA** starts processes by assigning high and low quality regions, in other words before processing each and every sequence like trimming 5’-3’ regions, it discerns and can totally ignore a sequence till later steps of contig building. When needed these low quality sequences are called and used in assembly, but with a very low frequencies (Bastien Chevreux et al. 2004; Bastien Chevreux 2005). Having fewer but more reliable reads left, **MIRA**next takes different clone template insert sizes and uses this information for a pre-assembly coverage analysis (Bastien Chevreux et al. 2004; Bastien Chevreux 2005) (Venter et al. 2001).

Employing a graph algorithm to locate potential overlaps, **MIRA** then uses dynamic programming to confirm the overlaps (Bastien Chevreux et al. 2004; Bastien Chevreux 2005). Having a pre-constructed graph and possible overlaps, **MIRA** searches the space and aligns read pairs, and starts contig construction processes, which are repeated many times (at default 3 times, but user defined) at any of the steps a low quality labeled read can be used to extend a contig’s length (Bastien Chevreux et al. 2004; Bastien Chevreux 2005). Before outputting the contigs, **MIRA** internally corrects some errors by comparing the repeats, tagged at sight, to already assembled contigs (Bastien Chevreux et al. 2004; Bastien Chevreux 2005).

1.2.1.1.4. Newbler (GS de novo Assembler)

Newbler has been developed by instrument manufacturer Roche Diagnostics itself, when they first announced the method for sequencing by synthesis using a pyrosequencing protocol (Margulies et al. 2005; Wheeler et al. 2008). The program's design is specifically for data generated by 454 (Roche Life Sciences) type sequencers (Margulies et al. 2005). It is fully compatible with data generated by 454 sequencers, can capture all information supplied by the instrument, including flow-based signal trace, which are made available with 454 technology (Barbazuk et al. 2007; (Margulies et al. 2005).

Newbler imitates the **Celera**-like process flow; starting with removing low quality regions. But having adapter sequences already tagged in 454 output data, it is much simpler than the same step in **Celera** (Genomics 2004b; Margulies et al. 2005). Having found high quality reads to process, this step is followed by an overlap finding step which performs complete all-against-all read comparison, which also simultaneously marks repeats (Margulies et al. 2005).. Having overlaps and repeats identified, **Newbler** starts assessment of read similarities by directly comparing flowgrams of each read (Margulies et al. 2005). **Newbler** introduces a specific choice at this stage; it takes the possibility of reverse complement reads into account. This feature gains **Newbler** the advantage of using more reads than many available assemblers (Barbazuk et al. 2007; Kumar and Blaxter 2010a; Liu et al. 2012), (Margulies et al. 2005; Lin et al. 2011a; W. Zhang et al. 2011; Cancer 2005).

Merging of the reads, contig construction, starts and is optimized in cycles. Contigs can extend or shorten on both sides, if supporting information is available. Size differentiation can occur in the presence of a repeat on the merge point of two reads, or if a supporting read is present in the read space, etc.

1.2.1.1.5. Phrap

Phrap (http://www.phrap.org/phredphrap/phrap.html) was originally developed for the assembly of cosmid sequencing within the Human Genome Project and made available in 1996 (Green 1997). It is the first assembler that on top of using quality scores generated by sequencing process, calculates base calling quality and uses this in the assembly process (Narzisi and Mishra 2011; Kumar and Blaxter 2010a; Green 1997). This feature helped the progress of identifying and assembly frequent imperfect repeats; bad random ambiguities-repeats occurring because of the sequencing processes' nature and existing real repeats like different copies of the Alu sequences in human genome (A. L. Delcher et al. 2002; Green 1997). In addition to its unique and effective solution to repeat finding, **Phrap** also is the first assembler to classify chimeras, sequencing and cloning vector sequences and low quality regions (Margulies et al. 2005).

Employing a greedy algorithm, **Phrap** starts merging reads that have overlaps of higher score and ends up with the read that has lowest scored overlap (Green 1997; Gordon, Desmarais, and Green 2001). Since **Phrap** is bundled with a finishing tool consed, contig quality assessment and editing are not performed (Green 1997; Gordon, Desmarais, and Green 2001).

1.2.1.2. Finishing Tools

Finishing tools are tools for further analysis preparation; performing processes like converting, filtering, viewing, editing, and finishing sequence assemblies by finding regions representing higher importance (higher coverage etc.) created with an assembler. Finishing capabilities allow the user to pick primers and templates, suggesting additional sequencing reactions to perform, and facilitating checking the accuracy of the assembly using digest and forward/reverse pair information (Bastien Chevreux 2005; Bastien Chevreux et al. 2004; Venter et al. 2001; Margulies et al. 2005).

1.2.2. Major Challenges of Assembly Processes

The major challenges of the assembly process basically come from the structure of genetic material itself: DNA has repetitive properties. Repetitive sequences are individual sequences that are similar or identical to sequences elsewhere in the genome. Repeats might originate from a number of biological mechanisms, and may comprise from a few to millions of copies and vary in size ranging from a few to millions of bases (Treangen and Salzberg 2011).

Repetitive sequences are observed in many organisms, including single cell bacteria to higher eukaryotes (Treangen and Steven L. Salzberg 2011; J. Jurka et al. 2007). It is claimed that this high level of repetitiveness is one of the major causes of bigger plant genomes in size: for instance, transposable elements cover more than 80% of the maize genome (Schnable et al. 2009). It also has been reported that half of the human genome is composed of these repetitive sequences (Treangen and Salzberg 2011).

These repeats are most probably the greatest challenge of assembly processes (Pop 2009a; Ramos et al. 2011; Paszkiewicz and Studholme 2010; Scheibye-Alsing et al. 2009; Schuster 2008b; Zerbino and Birney 2008a; Steven 2002a; W. Zhang et al. 2011; Lin et al. 2011b; Palmer et al. 2010; Cancer 2005; Metzker 2010); since sequencing yields errors and the shotgun sequencing method, especially when high coverage rates are desired, suffers from high error rates especially in homopolymer repeats (Z. Li et al. 2012; Scheibye-Alsing et al. 2009; W. Zhang et al. 2011). Thus, individual raw reads can have errors because of sequencing errors, but when sequencing errors are eveneliminated, a read might include polymorphisms that cannot be identified by assemblers, and marking those two reads as chimera-repeats, hence changing the course of assembly by complicating recognition of overlaps and contig constructions (Baker 2012).

Having repeat recognition and repeat treatment challenges overcome, comes another difficulty of assembly processes: the massive raw data that can be hundreds of millions of bases, 2.25 million in our study. Sequence assemblers must adapt to new technologies, must take advantage of whatever is offered by sequencing process and tolerate the faults, seek alternatives. But Wang and Jiang showed that (L and T. 1994), even when error free representation –reads and quality information - of true sequence is available, the assembly problem is NP-Complete. Meaning there is no such solution to be delivered fast in a feasibile way. This also means that assemblers should and would use approximation strategies, employing heuristic algorithms.

1.2.3. A Short Comparison of Competitor Sequence Assemblers

While overlap computing is definitely one of the very critical steps of an assembly, and each tool has its own solutions by means of employing powerful and accurate algorithms and setting a valid overlap threshold, competitor assemblers are using different definitions for a valid overlap (Table 3). Apparently every developer of the program has own explanation, but it really changes the outcome of an assembly (Pop 2009b; Narzisi and Mishra 2011; Bastien Chevreux 2005).

These assemblers differ in the algorithms they employ but overall they all use an OLC (**O**verlap-**L**ayout-**C**onsensus) algorithm, with very varying inner steps (Bartels et al. 2005; Lin et al. 2011b; Kumar and Blaxter 2010b; Bastien Chevreux 2005; X Huang and Madan 1999; Genomics 2004a). These differences have huge effects on how they treat individual reads; while some of them ignore the read that has been marked to be low quality and do not use it in the merging step, some can split the read from desirable position and can ultimately merge with reads or place into different contigs (W. Zhang et al. 2011; Bartels et al. 2005; A. L. Delcher et al. 2002; Z. Li et al. 2011).

There are many features for each assembler we took into consideration but Table 3 summarizes some to have an insight.

Table 3. A short feature comparison of sequence assemblers in scope. (Source: (Bastien Chevreux; Schatz 2006; X. Huang 1999; Margulies et al. 2005; Green 1997; Lin et al. 2011b))

	CAP3	Celera	MIRA	Newbler	Phrap
Valid overlap	40+/50	37+/40	24+/28	26+/30	N/A
Take advantage of Trace Info	No	No	Yes	Yes	Yes
Overlap calculation	SW	SW-Iterative	SKIM-Iterative	SW-Iterative	SW
Contig rebuilding	Yes	Yes	User defined	User defined	Phred*
Min. Read # for each contig	2+	2+	3+	3+	3+
Repeat recognition	**Consistency**	**Consistency**	**Consistency-Template**	**Consistency-Template**	**Consistency**
Recognition of uniform read dist.	No	Yes	Yes	Yes	Yes
Spoiler detection	No	No	Yes	No	Yes
Genomic pathfinder	No	No	Yes	No	No
Relative score	N/A	N/A	80+	60+	60+
Mark gap bases	No	N/A	Yes	Yes	Yes

1.2.4. Efforts Put Forth for Assessment

There have been many studies performed in order to find answers to the de novo assembly quality assessment problem. Assemblethon (http://www.assemblathon.org), dnGASP (http://cnag.bsc.es/), GAGE (http://gage.cbcb.umd.edu/) can be listed as the top collaborative efforts.

The Assemblathon (**Assembl**y Mar**athon**) evaluated almost a hundred metrics of sophisticated statistics in terms of how complete and accurate the assemblies were, by taking advantage of 41 assemblies generated by 23 independent sequence assembly tools from 17 institutes around the world (Earl et al. 2011).

Assemblathon was organized by experienced and well-known scientists in the area of sequence assembly and genomics. Rather than use an existing reference genome’s real data generated from next generation sequencing technologies, they took Human ch13, divided into 4 chromosomes, contaminated it (5%) with E.coli sequences, and simulated this data as if it has been evolved for ↨~200 million years. They repeated the evolving step by diverged into two independent lineages, both simulated to evolve for more ~50 million years, both having the same size genome. This way they aimed to have genome orientations randomly and prevent simple discovery of the un-biased data. These two lineages were sent to teams who were asked to perform de novo assembly (Earl et al. 2011).

Analyzing the result they tried to come up with statistical metrics, that will unveil the right assessment criteria. Though N50 did correlate roughly with assembly quality, they concluded that no set of metrics was perfect.

The overall conclusion was that conservative assemblers which were adjusted for earlier sequencing technologies, require extensive overlaps and robust data in terms of quality and length, to join reads into contigs. Aggressive assemblers, that have been announced recently, can work with hundreds of millions of reads, produce longer contigs accurately, cover the entire genome, but are more likely to join regions in the wrong order or orientation (Baker 2012; Sahli and Shibuya 2012)

1.3. Aim of the Study

As mentioned many times in the introduction, demand for next generation sequencing is increasing and with a very promising proportion. Since the development of these revolutionary technologies that deliver fast, accurate and inexpensive massive genome information, most biological and biomedical applications fundamentally shifted away from conventional methods like Sanger sequencing to these technologies (Margulies et al. 2005; Metzker 2010; Pareek, Smoczynski, and Tretyn 2011). This obviously means the use of sequence assembly programs is also rising (Paszkiewicz and Studholme 2010; Jaffe et al. 2003b; Bastien Chevreux et al. 2004; Salzberg and Yorke 2005; Margulies et al. 2005). The increase of usage of these programs is also visible from the increasing number of announced sequence assemblers (Bastien Chevreux et al. 2004; Lin et al. 2011b; W. Zhang et al. 2011).

While ease of acquisition of data is always desired, accumulation of data brings another problem; these new sequencing technologies also pose tremendous challenges to traditional de novo assembly tools designed for conventional sequencing techniques, as they are incapable of handling the millions to billions of short reads and incapable of taking full advantage of ancillary information that these new technologies provide (Z. Li et al. 2012; Liu et al. 2012; W. Zhang et al. 2011; Narzisi and Mishra 2011).

When a genome study of a previously unsequenced organism is revealed, the first question asked by all is: “is the outcome accurate-reliable?” (Salzberg and Yorke 2005; Jaffe et al. 2003b; Huse et al. 2007; Myers et al. 2012). The genomicists claim the accuracy of the accomplishment with the experiments they perform but in the absence of a reference genome it is really difficult to be sure of the accuracy (Baker 2012; Kumar and Blaxter 2010b; Lin et al. 2011b). This outcome is definitive from the effort of the most commonly studied completed genomes such as human and mouse. For example, in 2011 Alkan and colleagues found that a de novo assembly of the human genome was missing 420Mb repeated regions and over 2.000 protein-coding exons (Baker 2012; Narzisi and Mishra 2011, Alkan, Sajjadian, and Eichler 2011)

Some studies (Istrail et al. 2004; W. Zhang et al. 2011; Z. Li et al. 2012; A. L. Delcher et al. 2002; a L. Delcher et al. 1999; Liu et al. 2012; Kumar and Blaxter 2010b; Lin et al. 2011b) focused specifically to the performance of these tools under various conditions, and shed more light on the possible drawbacks and limitations of assemblers. However, the accuracy and efficiency of these tools, more precisely the criteria of better assembly is still frequently discussed (Baker 2012, Quail et al. 2012, W. Zhang et al. 2011) and yet has not been fully investigated. Sufficient information is not currently available for informed decisions to be made regarding the tool that would be most likely to produce the best performance under a specific set of conditions (W. Zhang et al. 2011; Z. Li et al. 2011). It is a non-negligible necessity to systematically analyze their relative performance under various conditions, not just the assembly statistics like broadly mentioned contig size, length etc. but comparison with experimentally validated data like EST-cDNA sequences, so that researchers can select a tool that would produce optimal results according to their specific requirements (Materials et al.; B Chevreux; Sahli and Shibuya 2012; Liu et al. 2012; Z. Li et al. 2012; Istrail et al. 2012; Baker 2012). Using transcriptomic data is crucial, for some (Baker 2012) every genome project should have a parallel transcriptome project- to identify the intron-exon structure within genes and aid scaffolding and annotation processes (Morin et al. 2008; Weber et al. 2007), (Baker 2012; Cancer 2005), (Salzberg and Yorke 2005; Pop et al. 2004).

This study aims to compare the performance of the aforementioned sequence assembly tools by means of:

- Accuracy and efficiency; not only with commonly addressed statistical metrics like contig length, size, and cumulative base length of assembly, but also investigating the possible left overlaps in the assembly, to determine if the unused reads are correctly adjusted
- Validity and effectiveness in biological context; by the introduction of experimentally validated data like mRNA sequences and primer sequences, and understand at what proportion assemblies can compensate.

CHAPTER 2

MATERIALS AND METHODS

2.1. Environment

All of the processing and calculations of this study were performed on a workstation that has 4 core- 8 thread CPU running at 3.8 GHz, 48 GB of RAM running at 1.6GHz and two SSD drives installed as an RAID-0 disk array for operating system and 3TB of HDD for data storage. As an operating system a Linux distribution Ubuntu was chosen, with kernel version 3.2.1. Multi threads and the RAM capacity of the workstation let us perform multi-task processing simultaneously. Observing the amount of the memory usage of the assembly tools, having this amount of memory is recommended, yet not an obligation. SSD drives warranted us a domain to work with hundreds to thousands of independent files without coming across a bottleneck.

2.2. Raw Data

There are 4 datasets used in this study: two are raw data originating from 454 FLX sequencer that was made available from studies performed by Çelik (Çelik 2011) and Tekin (Tekin 2011). These two were the datasets that have been input to the sequence assemblers. The statistics of the genomic data are given in Table 4.

Table 4. Raw data sets (genomic) generated by 454 sequencers

	Papaver somniferum L	Sesamum indicum
Read count	1.244.412	1.094.317
Min read length (b)	40	40
Max read length (b)	764	900
Avg read length (b)	380,4	348
SD (b)	142,5	125,6
Total base count	474.398.321	380.862.690

In order to have supplementary information on which assembly result will project more of the real sequence or the organism, we chose to use EST- cDNA dataset of the organism (Papaver somniferum L.), from publicly available databases (NCBI). In order to discriminate and bypass the clutter that can be caused by another species of the same genius, we filtered the data with the taxonomy ID of Papaver somniferum L (txid:3469).

Table 5. Raw data set fetched from public databases

	Papaver somniferum L
Read count	21.330
Min read length (b)	18
Max read length (b)	216.086
Avg read length (b)	848,6
SD (b)	2300,3
Total base count	180.994.437

The second data for supporting the originality of the assemblies were a number of unique primer sets; ranging from 18-24 bases long, that wereexperimentally validated to belong to the same organism which its DNA has been sequenced (Morin et al. 2008).

2.3. Read Pre-processing

Roche 454 instruments –GS, Titanium, FLX- outputs raw reads in **S**tandard **F**lowgram **F**ormat (SFF) files that include base quality and clipping information, and additional information such as flow based signal traces, etc. (Margulies et al. 2005; Cancer 2005). In order to claim an objective comparison set forth, we ensured that each assembler got exactly the same input, so we removed 454 adapters that were used in the sequencing process from the raw reads ourselves prior to assembly process; rather than let assemblers use their own built-in adapter removal tools. This is essential since **CAP3** and **Phrap** do not remove adapters at all, but **Celera**, **MIRA** and **Newbler** pre-processes reads slightly differently (W. Zhang et al. 2011; Kumar and Blaxter 2010b), the use of a dataset free of adapters will ensure a fair head-start for each assembler.

SSAHA2 (http://biowulf.nih.gov/apps/ssaha2.html) was used to remove the contaminants; since it already has a built-in compatibility feature with next generation sequencing technologies –we used -454 switch. Because the adapters used for sequencing were known and provided with the raw data, contaminant removal was done swiftly.

Having contaminants removed from the data, we used **sff_extract** (http://bioinf.comav.upv.es/sff_extract/) of COMAV to convert the default file format to a more common file format that every assembler can work on: **FASTQ**. **FASTQ** (**FASTA** and **Q**uality) is a file format that can store both the biological sequence and its corresponding quality information at the same time.

2.4. Assembly

Every assembler's work was monitored with our own in-house pipelines; the runtime, CPU, memory and disk usage, etc. For the sake of fairness, only one assembler at a time was run(Table 6).

All assemblers were set to use default parameters, so to have the absolute indicator of each and every the assembler.

2.5. Mapping

We used two different types of data, both in **.FASTA** format. The first data consisted of EST-cDNA sequences which tend to be in exonic regions of the genome. Since our genomic assemblies might cover intronic and exonic regions at the same time, a program that will tolerate and adapt to this situation was chosen: **BLAT** (**B**LAST **L**ike **A**lignment **T**ool) (Kent 2002). Another important reason that directed us to choose **BLAT** is, unlike **BLAST,** it indexes the databases into the memory, since memory and hard drives have an incomparable speed difference (~80MB vs ~12.000MB per second), it gained us a lot of speed.

2.5.1. Residual Overlap Calculation

BOWTIE (http://bowtie-bio.sourceforge.net/index.shtml) is an aligning tool which especially aligns short DNA sequences with high efficiency to the genomes (Langmead et al. 2009). We used **BOWTIE** in our residual overlap calculation analysis.

Residual overlap calculation is a method that consists of the analysis of the contigs within themselves; performing pairwise alignments all-against-all, in order to understand if there are overlaps that could be a potential merging point of two contigs. All assemblers analyze all sequences and try to merge those with enough evidence to be consecutive, to a new sequence of a greater length. That merging process of course can be done by having the overlap point and sizes -the valid overlaps-.

Residual overlap calculation might reveal information especially on why tools have different settings for a valid overlap, and more importantly what would be a safe threshold for a valid overlap. If valid overlap calculation is done with underset settings, that might end with the assembler performing aggressive joining of sequences, generating longer contigs.

From each and every assembly we filtered sequences with the minimum length of 300 bp. This threshold was not selected arbitrarily. Of the assemblers under examination, **CAP3** has the highest set threshold for default overlap calculation. It takes at least 50 bp of a sequence from left and the right regions, and seeks 40+bp to be similar to an independent sequence’s left/right regions.

From contigs longer than 300 bp, the first and last 100 bp were cut, marked as to know the origin sequence. Those fragments were then pairwise aligned all-against-all and a percentage of hits was calculated for each assembly.

2.5.2. Mapping Singlets to the Contigs

Assemblers among many difficulties have to try and identify the sequence list, that will be used in the assembly and beware of a number of sequences that might not be beneficial if used in the assembly. With next generation sequencing technologies providing millions of sequences, this step is really challenging for an assembler.

If an assembler falsely selects not to use a sequence and keeps it as a singlet, this might very well be a critic piece of a great puzzle. This might result in not being able to construct a longer sequence that might represent a greater and more accurate projection of the real sequence, and so on.

For the analysis, we started with removing the debris sequences from the outer contigs data, keeping only singlets. Since the shortest raw sequence length was 40 for both our genomic datasets, we filtered sequences based on this criterion. This step was necessary, because some tools output debris sequences falsely within the same pool of singlet sequences. This step in general dropped the total singlet sequence count by 4%. After this step, we mapped all singlets to contigs. Mapping results were deposited to a **MySQL** database, with about 15 million rows, for each assembly. In order to be precise about whether a singlet could really be used in the assembly we performed a second filtering with sequences that have 90% similarity to a contig. Satisfying this criterion left around 80 thousand singlets. This filter was not enough to reveal a singlets' situation to be a "proper" sequence, which could be exploited. This is because assemblers tend to exclude a sequence if it is similar to many other sequences, making it a potential repeat. Assemblers also tend to ignore sequences which consist of many repeated regions within itself. So in order to overcome those difficulties and avoid the removal of accurately recognized singlets from the analysis, we filtered sequences that have many hits to a contig and 60+% repeated regions. This time we had around 35 thousand of rows in our database.

2.5.3. Mapping Primer Pairs

We have used a set of primer pairs that were designed from the same sequencing data for an earlier study. Out of many primer pairs, 50 were selected and experimented in wet lab. Amplification test of those primers validated the primer pairs and product sizes were measured.

Since a primer pair maps to DNA sequence as forward primer to 5' and reverse primer to 3', we opted to locate primer pairs relying on this criterion. Having the real product sizes of the mapped primer pairs in hand, a second analysis was performed. The part of the contig from the beginning point of the forward primer to the end point of the reverse primer was cut out and compared to experimental results. This step revealed information about the accuracy of an assembly, since a primer pair usually amplifies only one locus and product sizes are known.

2.5.4. Mapping EST- cDNA Data

EST sequences are short subsequences of complementary DNA. This information is the sum answer to the analysis. The optimum assembly might cover more EST sequences; even the pieces of EST sequences. We mapped all ESTs to assemblies separately; keeping in mind some EST sequences might be longer than contigs.

Mapping resulted in at least 10 million hits for each assembly, which we again organized in a **MySQL** database. To get the brief knowledge we started processing the data by first summing all hits of a specific EST to the same contig, dropping row count in the database by a million on average. Then we filtered the contigs, assigning an EST each, with the longest EST hit, dropping database rows to about 50 thousands. Finally we filtered EST sequences to indicate only one contig with the longest hit, finalize the database with row count of around 9 thousand.

Having unique ESTs identified, we calculated the contigs' EST hit coverage by first getting all EST hits to the same contig, which was paired with the unique EST. We referred to the largest unfiltered database with all hits included, fetching the hit size, hit start and end point of an EST sequence. We generated an array of length the contig, called **contigArray**. Starting from the pristine EST hit, we incremented the indices of the **contigArray** by one beginning with pristine EST's start point to the end point. With the second EST hit, we started the same index incrementing procedure and this step was performed as many times as there were EST hits present for the same contig. The coverage value for a contig was calculated as the sum of the greatest indices. The method is demonstrated in Figure 3.

Having a coverage value for each and all contigs calculated, we compared assemblies with respect to cumulative coverage counts, min-max and 1st-3rd quartile lengths and presented the total coverage results.

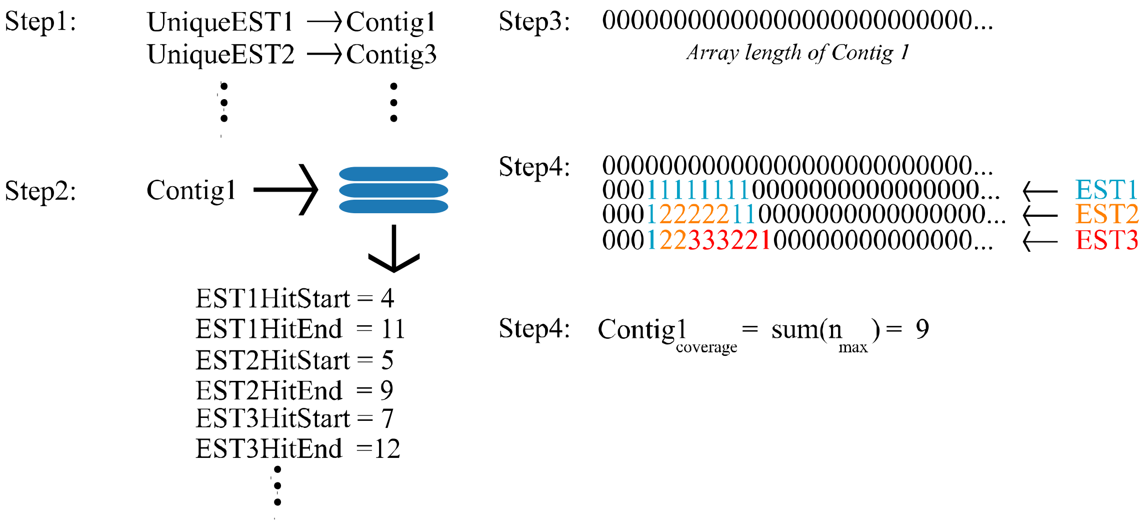

Figure 3. The process of calculation of the coverage for contigs.

2.5. Other

Since the amount of data and inner steps were many and to be repeated many times, we developed pipelines to speed the processes and avoid manual errors. All the tools were programmed with **Java**, and all the scripts were written with **bash** and **Perl** languages.

2.6. Quality Assessment Criteria

The optimal assembly should project the following:

- The examination of left overlaps of an assembly should reveal that no or very few overlaps are present in the assembly
- The examination of locating unused reads in the assembly step should reveal the maximum values, representing that the assembler took advantage of reads that could be part of assembly optimally and ignored repeated regions accurately.
- The examination of mapping cDNA sequences to the assembly step should reveal maximum results, that the optimal assembly covered more cDNA sequences than is believed to be present in the genome.
- The examination of mapping primer sequences that weredesigned from the same organism’s genome and experimentally validated to be in the genome, should reveal maximum results, that more primer sequences could be covered with the optimal assembly.

CHAPTER 3

RESULTS

3.1. Statistical

Tools were run on an idle workstation once at a time, assuring all calculation power is available for each. Table 6 repsesents hard drive, memory and process consumptions of each assembler and runtimes.

Table 6. Real time process consumption of the assemblers (the poppy assembly only)

	CAP3	Celera	MIRA	Newbler	Phrap
Memory usage (peak)	24 GB	33 GB	16 GB	11 GB	22 GB
Runtime	1.274 minutes	1.543 minutes	822 minutes	734 minutes	1.457 minutes
Threads used	1	1	2	2	1
Disk usage	2 GB	5 GB	24 GB	4 GB	6 GB

Although the most commonly used metrics to represent an assembly's quality are contig length and contig count, contig length is not an exact indicator of accurate or inaccurate assembly (B Chevreux; Liu et al. 2012; Baker 2012), (Parra et al. 2009). Because during the assembly processes an assembler could simply mark a whole read as repeat, hence ignoring it fully. The optimal assembler should avoid building over-assembly of reads into in silico chimaeras rather than expected contigs, and avoid the production of near-identical, largely overlapping contigs from allelic copies or error-rich data in the contig building step (Istrail et al. 2012; Lin et al. 2011b; W. Zhang et al. 2011; Parra et al. 2009; Z. Li et al. 2011; Margulies et al. 2005).

Figure 4 demonstrates the contig length statistics. **MIRA** outputted the longest contigs for two organisms by showing a consistency that indicates **MIRA**'s aggressive read merging step for the sake of longer contigs. In terms of consistency, **Phrap** also outputted the longest contig of size ~100Kb.

CAP3 outputted the longest contig size compared to other assemblers, very short. In fact the expectation from the **CAP3** was to output much longer contigs than it did, and compared to other assemblers to hit the average (X Huang and Madan 1999). However taking into consideration that **CAP3**'s shortest outputted contig length was bigger than the other assemblers result, it raises suspicions to whether **CAP3** can determine the low and high quality reads as it should.

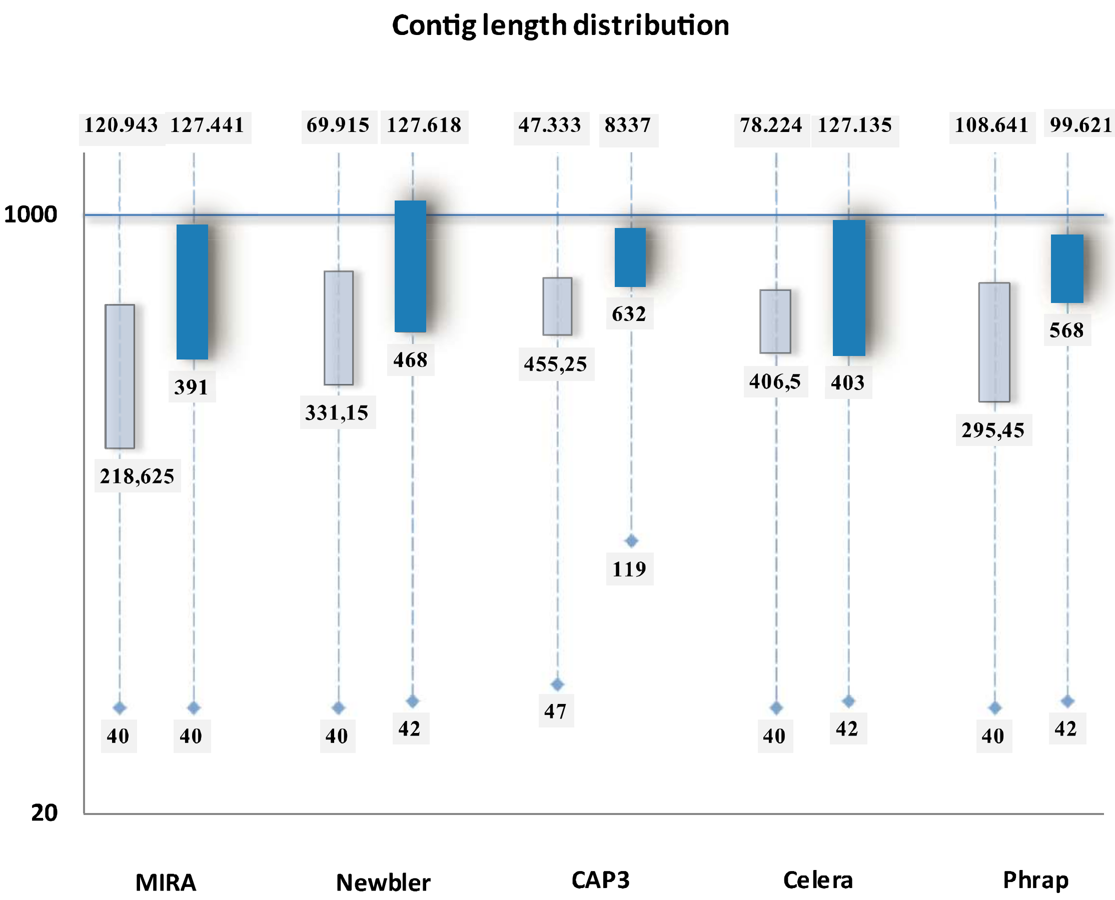

Figure 4. Minimum, maximum, first and third quartile length of the contigs from 5 individual assembly of two plants' genomes. Since the interval of min and max contig sizes is great to plot, even with the introduction of a $\log_{10}$ X axis, the chart was cut down and the peaks are not shown but the values are. Light blue represents Papaver assembly data, while dark blue represents Sesame assembly data.

Another metric that is mentioned together with the contig length is contig counts (Lin et al. 2011b; Z. Li et al. 2012; W. Zhang et al. 2011; Pop 2009a; Margulies et al. 2005).

MIRA outputted the highest contig count exceeding 73 thousand spanning ~62 megabases, which is 13% of the raw reads. The least amount of contig count was outputted by **Newbler** with 31 thousand, but spanning ~57 megabases that is 12% of the raw reads. Figure 5 represents contig sizes and contig counts.

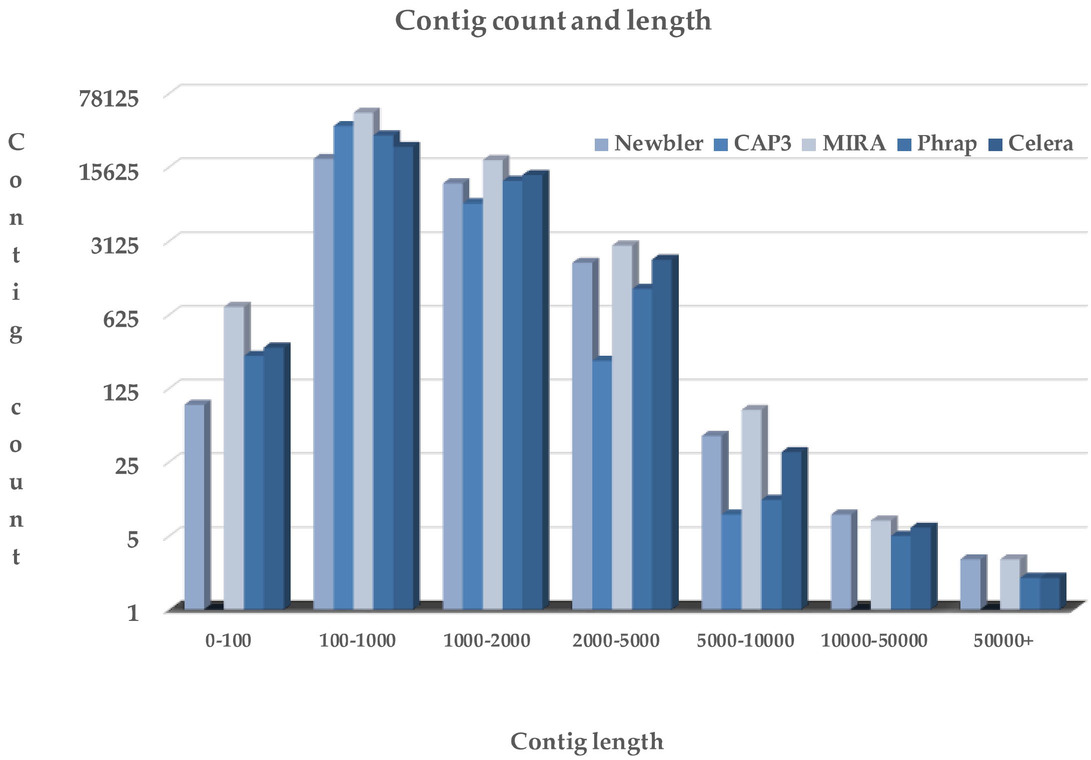

Figure 5. Contigs lengths clustered from 5 individual assemblies of Papaver somniferum L. genome.

Having contig lengths and counts might not clearly represent the assemblies overall size; in order to see that, a cumulative contig length comparison is given in Figure 6. For both of the organisms **MIRA** outputted the longest assembly where **CAP3** outputted the shortest.

Cumulative total assembled bases is not an exact indicator of accurate or inaccurate assembly, especially for de novo sequencing projects, in the absence of reference genome, longer assembled contigs could simply be overmerged reads resulting because of a false set for determination of a valid overlap (Bastien Chevreux 2005; Bastien Chevreux et al. 2004; Baker 2012; W. Zhang et al. 2011).

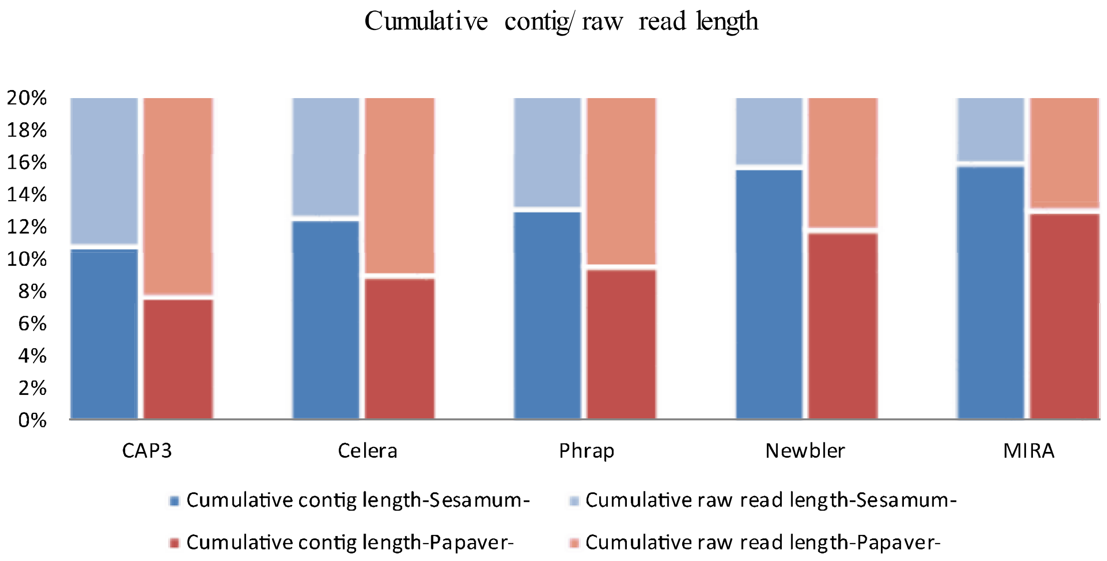

Figure 6. Cumulative contig lengths clustered from 5 individual assemblies of Papaver somniferum L. genome.

Figure 7 represent used reads of the raw reads in percentages. Used and unused read count percentages in an assembly while informative is not a valid metric itself, because of the heterogeneity of the raw data. Tagging a read as debris would result in an assembler not adding the read to the assembly pool, but this can simply be because there was not enough proof that the contig containing that read would represent real sequence (Istrail et al. 2004; Baker 2012; Margulies et al. 2005)

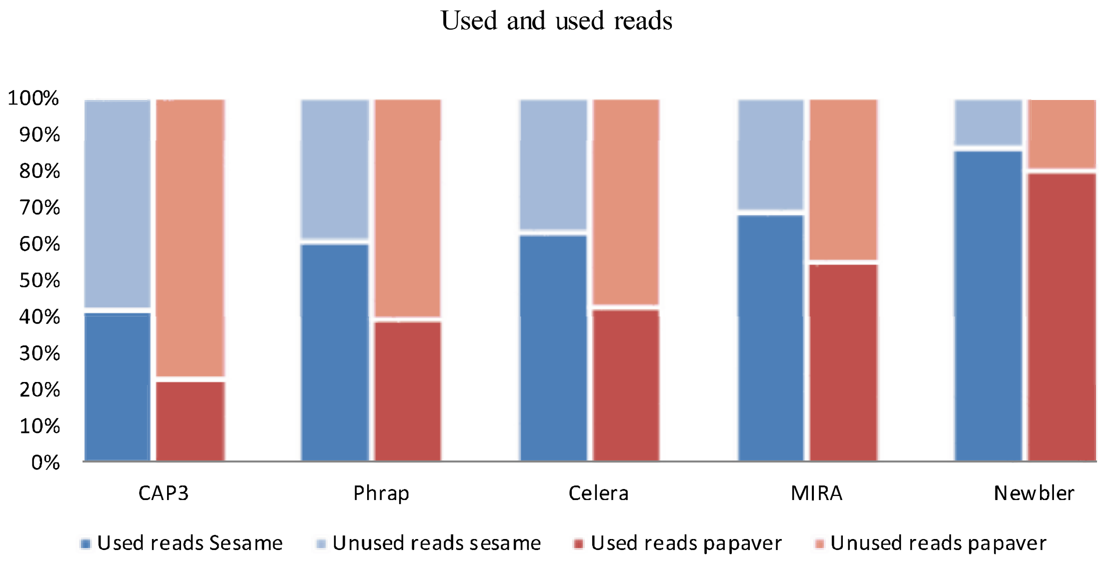

Figure 7. Used and unused read percents from 5 individual assemblies of two organism' genomic data.

To this point, basic statistics were displayed, but with the data presented thus far, it is still unclear why assemblers treat data so differently. For instance, while **Newbler** uses by far more reads in the assembly, it does not deliver the longest assembly in size. This simply can be explained by **Newbler** treatment of the reads; that it can divide reads and use those inner regions.

Figures 9 and 10 represents overall coverage; meaning evidence that assemblers use to get the valid overlaps, and merge the reads. Those coverage statistics were generated by our own script; taking advantage of the .**ACE** formatted file that contains supporting information like reads used to construct that corresponding contig, every base's quality scores, etc. We simply calculated the amount of read, read length that has been used to construct per contig and converted this into coverage information. Figure 8 illustrates the coverage calculation.

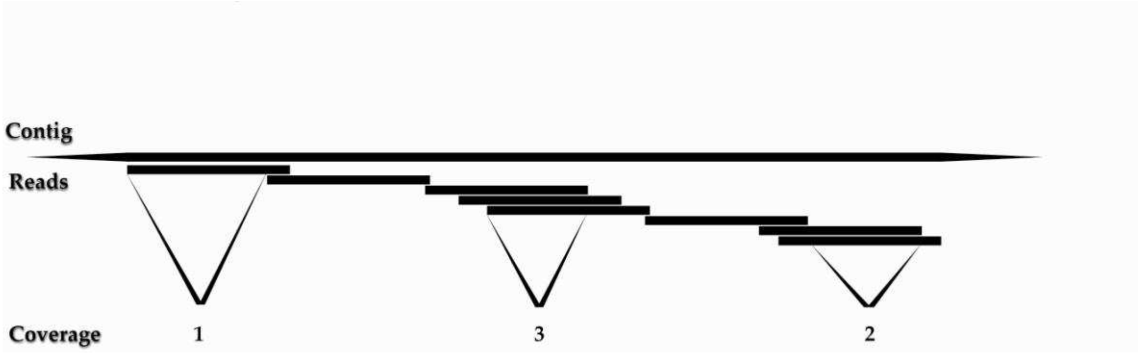

Figure 8. Coverage calculation is performed as follows: the sequences that constitute a contig are put into order with respect to their start positions, without leaving out the overlapping regions. Incrementation of all indices by one of contig size array begin with the first sequences in a row. Incrementation continues till the end of smaller sequence, and with a new sequence starts new incremention of the array indices. The inner indices that were covered by many sequences will be incremented many times while an index will be incremented only once if it has only one sequence at that region.

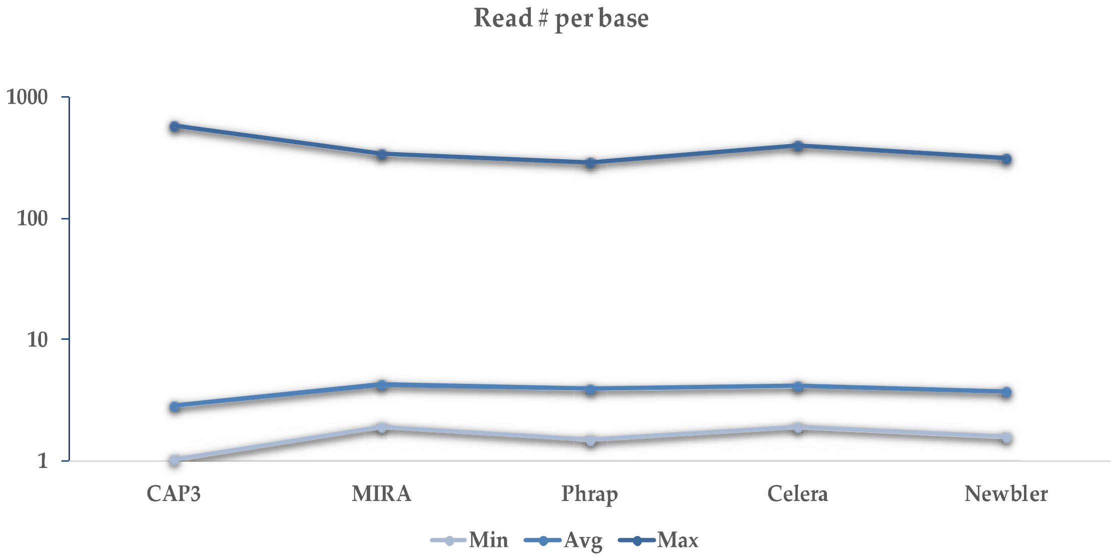

Figure 9. Coverage distribution of assembly results generated from 5 individual assemblies of Papaver's genomic data.

These represented results so far –excluding the coverage information- are supplied almost with every assembly study to provide quick information to the reader. But these statistics in reality do not reflect the whole. Since there is no reference genome available for de novo assembly statistics, researchers are pushed to rely on the reported data.

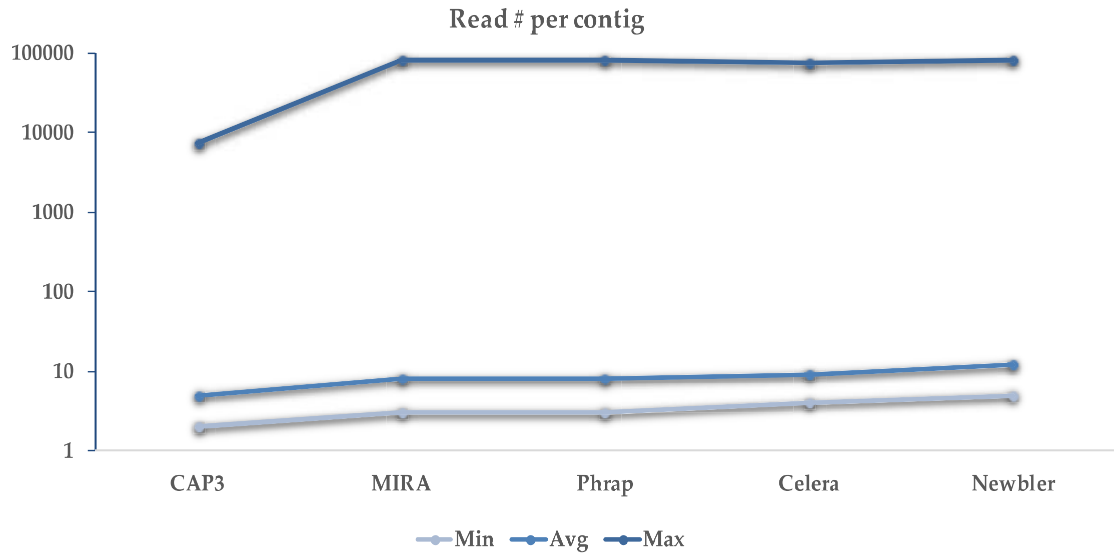

Figure 10. Coverage distribution of assembly results generated from 5 individual assemblies of Papaver's genomic data.

3.2. Mapping

In order to go beyond the knowledge the basic statistics provided, and more importantly to have an idea about the validness of the assemblies biologically, we opted to introduce a comparison step: assemblies vs. experimentally validated data that can assess the real outcome.

3.2.1. Residual Overlap Calculation

Results showed that for 454 sequencing data, the safe threshold should be no more than 30 bp from both ends. Of all the assemblers under examination, **CAP3** had the highest default set threshold with 50 bp, and results indicated that **CAP3** assembly results still include potential overlaps, much more than the other assemblies.

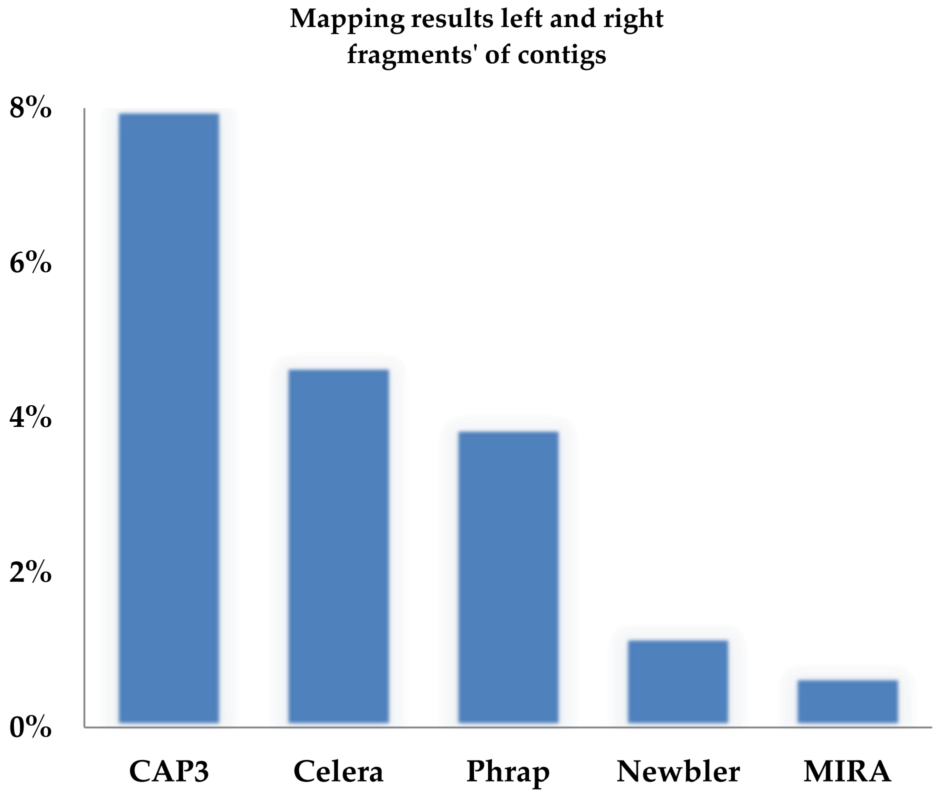

Figure 11. Mapping results of all assemblers for Papaver data. 8% of the contigs had potential merging points, but was not taken into consideration, in the CAP3 assembly. This might be the answer to why CAP3 could not output the longest contigs sizes, while other assemblers could.

3.2.2. Mapping Singlets to Contigs

Mapping singlets to contigs analysis is another aspect of how overlap calculation might affect the total assembly. If an assembler cannot identify a valid overlap between two sequences, it might not take advantage of those. Of course, as stated before although an overlap was found that sequence might not be put into assembly for several reasons like unresolved layout problems, multi match points etc. (Cancer 2005).

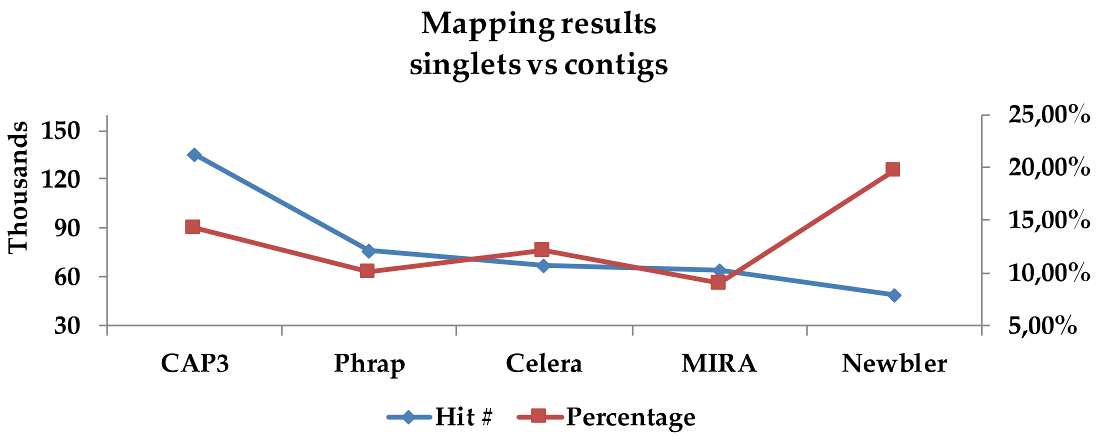

Figure 12. Mapping singlets to contigs plotted into two axis; total match count and percentage of singlets matching contigs, of all singlets.

Results showed that **CAP3** left more than 130 thousand singlets that could potentially change the course of the assembly drastically. From the statistical analysis it was shown that **CAP3** left around 900 thousand of the raw sequences out of the assembly, and took advantage of only about 300 thousand raw sequences. This is by far the highest rate of sequences kept out of the assembly.

When it comes to lowest rate of singlet matching to contigs, **Newbler** performed as a prime tool. It was the tool which was able to take advantage of the most raw sequences, kept the lowest rate of the reads out of its assembly, around 300 thousand. Mapping singlets to contigs analysis showed that **Newbler** could identify the singlets most accurately among all tools.

The percentage values in Figure 12 indicate the accuracy of each tool assigning a singlet as a real singlet. **Newbler** had the least amount of singlets after assembly, so the singlet-contig matches to all singlets ratio is the highest.

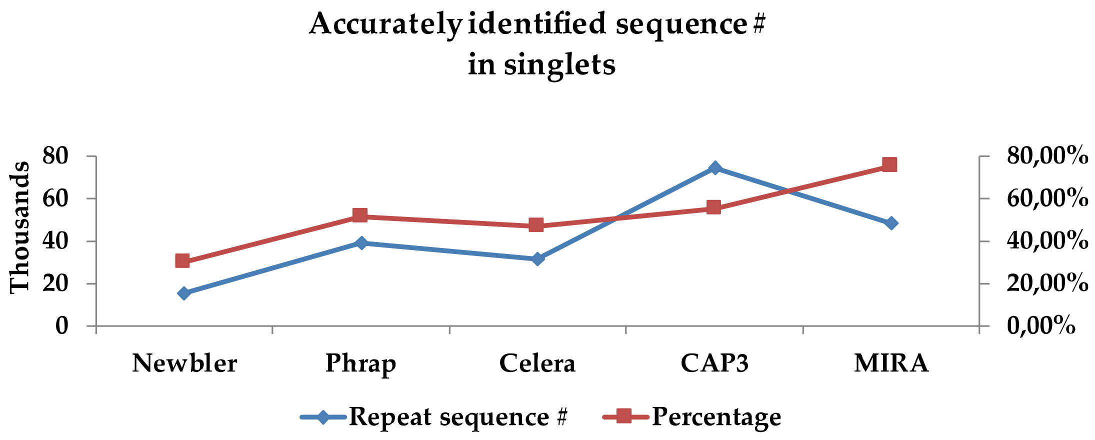

Figure 13. Overall 50% of the singlets in all assemblers were correctly identified. While MIRA seems to be the prime tool with identifying the true singlets, Newbler in fact performed much better by taking advantage of many more raw sequences into the assembly.

In order to truly reveal the information for each assembler, whether it could recognize the potential real singlets from the potential sequences which could be used in the assembly, the singlets with matches to contigs were further analyzed. Repeated inner regions, falsely called bases in the sequencing process (implanting Ns into bases where not enough evidence could be gathered) might be enough to tag a sequence as invaluable for assembly.

Of all assemblers, **MIRA** was the leading tool when it came to identify whether a sequence is a singlet. With more than 45 thousand accurately identified singlets, **MIRA** could correctly recognize more than 80% of the sequences in the pool of singlets matching contigs. Following **MIRA, CAP3** recognized about 50% of the singlets correctly. This situation brings the question “Why **CAP3** could identify singlets relatively well, but why it cannot take advantage of more raw sequences at the beginning of the assembly, and output this many unused sequences?” Having observed the **CAP3** outputs, it generally does not use parts of a sequence but tends to use it as a whole. This echoes the problem of setting a safe threshold for a valid overlap calculation.

3.2.3. Mapping Primer Pairs

Experimentally validated (amplified) primer pairs are good indicators of an assembly’s accuracy; since there is evidence that indicates the real sequence encapsulates the primers. Also wet lab experiments yield the product sizes for each primer pair, which can be expected to be satisfied by the optimum assembly.

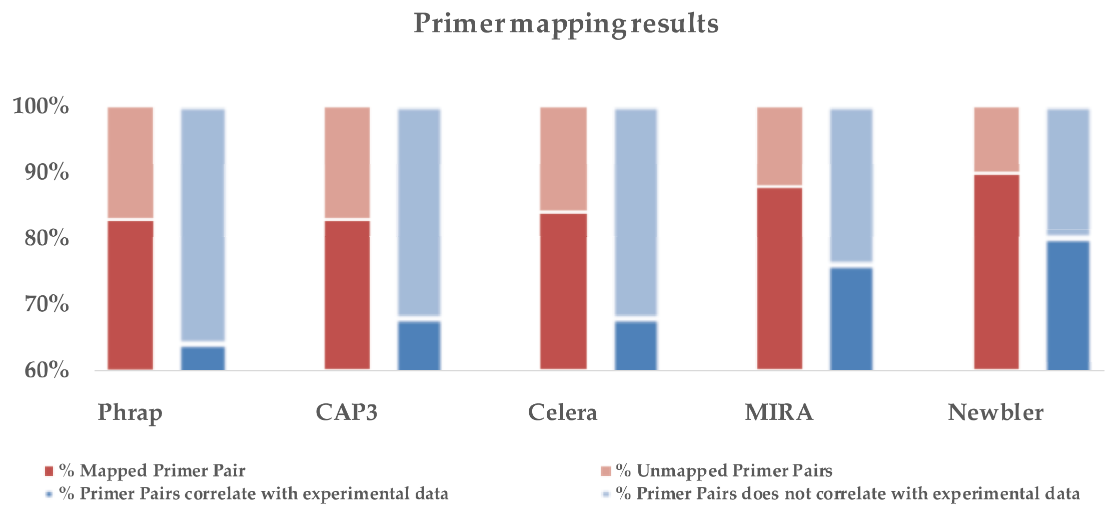

Figure 14. Primer mapping results plotted together; the amount of primer pairs mapping to contigs and primer pairs that satisfied expected product sizes.

Results indicated that all assemblies could provide a circumstance where more than 80% of the primer pairs could be mapped. But with this analysis, the decisive comparison is the comparison of expected and present product sizes’. We tolerated ±10% length deviance of expected product sizes before the calculation of the present product sizes. **Phrap** results showed less correspondence with expected product sizes, while **Newbler** showed the highest correspondence.

3.2.4. Mapping EST-cDNA Data

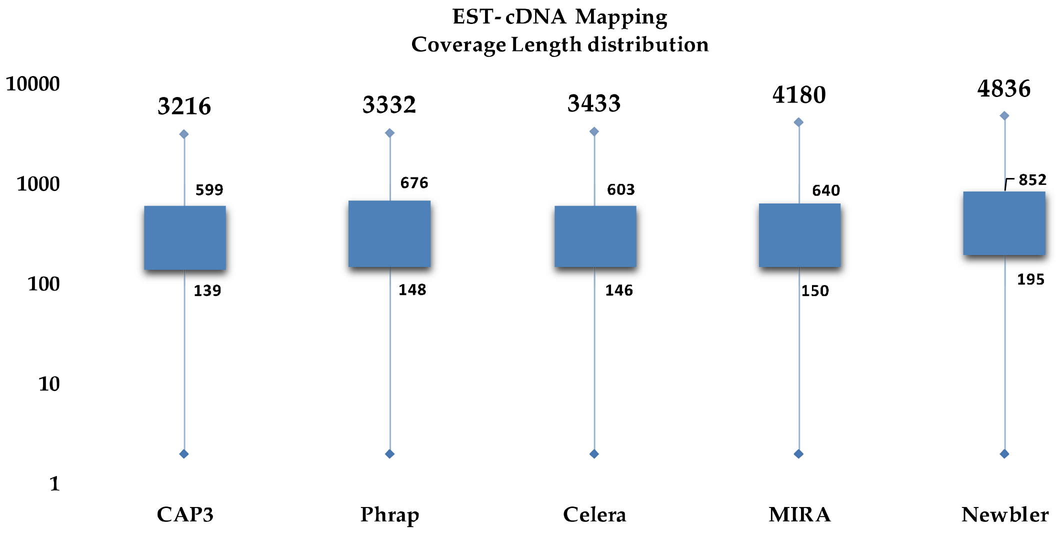

Figure 15. EST- cDNA mapping results plotted altogether as minimum, 1st quartile, 3rd quartile and maximum lengths.

EST- cDNA mapping results indicated that while **CAP3** assembly covered the least amount of EST- cDNA sequences, **Newbler** covered the most overall and it was 50+% more than provided by **CAP3**. This result could be expected only if **Newbler** assembly was the longest assembly by means of cumulative contig length, or if it had the highest mean contig sizes. But for those two metrics **Newbler** outputted results were behind what MIRA outputted. So Newbler assembly covered more EST- cDNA sequences because it presented the largest assembly, but it might be because it took advantage by far of the vast amount of raw sequences at the stage of assembly.

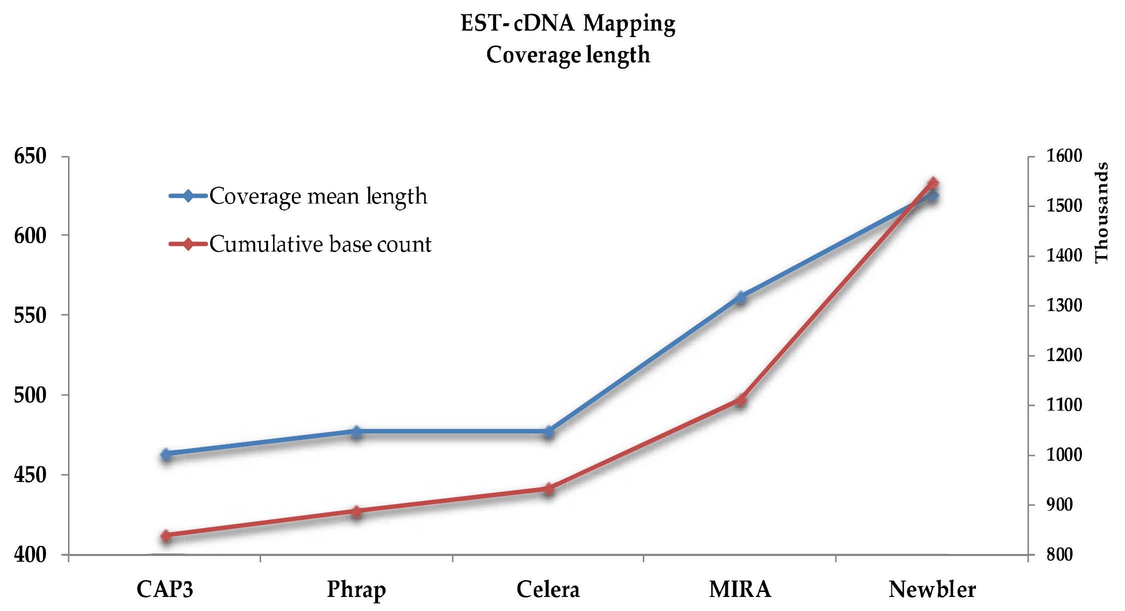

Figure 16. Mapping results of EST- cDNA sequences to each assembly.

3.3. Custom Assembly

After many analyses of the assemblies, we decided to fine tune the parameters with what we learned. For several reasons like; easier customization, rich features, being able to use the workstation resources to the end by means of taking advantage of all present calculation power and so on, we opted to proceed with **MIRA**.

Our workstation had 8 threads CPU, so we set **MIRA** to use all of the threads, hence speeding the processes.

First of all we set **MIRA** as not to clip the sequences, since we removed the adapter and linker sequences prior to assembly and also because the data we possess does not have enough coverage. **MIRA** has an option to run employed **SKIM** algorithm that will re-calculate overlaps for each pass, and we opted to redo the contig building steps 6 times. This is expected to overcome the misassemblies due to possible repeats. Having the mean size of our data, we set **MIRA** to take this value into consideration, saving some calculation time, especially at the stage of repeat recognition.

We set **spoiler detection** to run for each pass**,** which runs at the stage of repeat recognition and contig rebuilding. A spoiler can be either a chimeric read or a read that has unclipped long vector sequences still included, which in the end causes contigs not to be merged, or not to take advantage of the entire sequence. We also set **MIRA**'s internal **genomic pathfinder** algorithm, which aids genome building processes.

We also forced **MIRA** to mask the repeats that are found more often than the median occurrence of common repeats thus aiding repeat recognition steps. **MIRA** has an embedded contig editing function, which we used, that is capable of recognizing the homopolymers, one of the deficiencies of 454 sequencing technology.

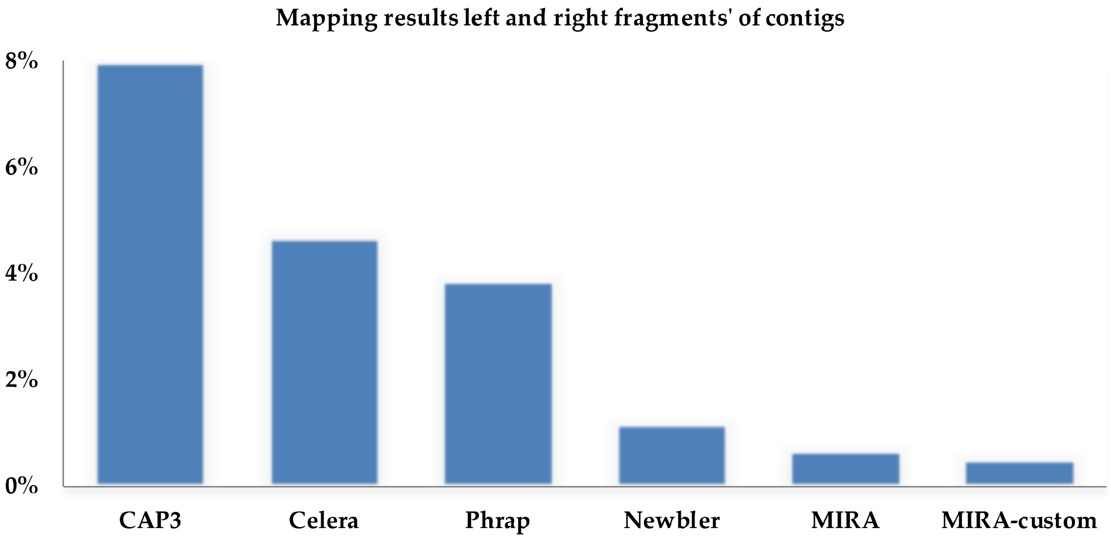

Figure 17. Results that include the latest custom assembly of residual overlap calculation analysis. MIRA had proved its quality by means of the calculation overlaps efficiently, new setting made MIRA more sensitive and produced better results.

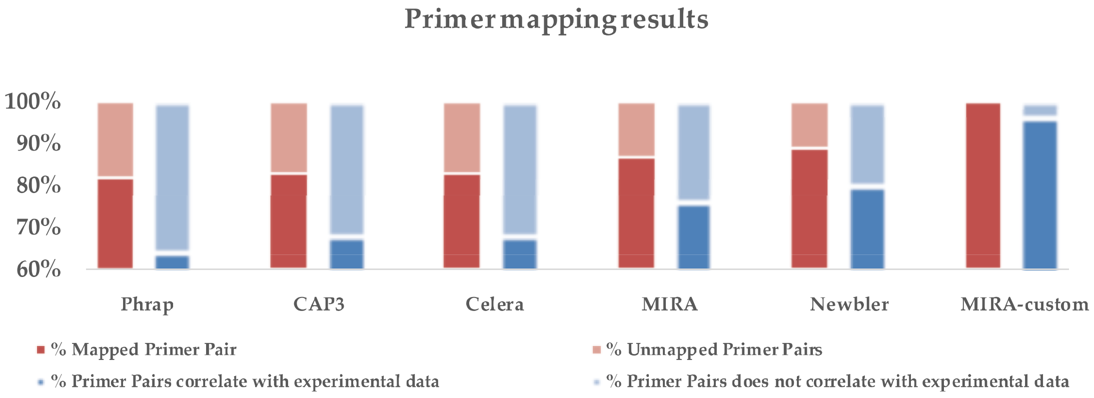

Figure 18: Mapping results of primer pairs to the assemblies.

Mapping primer pairs to the custom assembly resulted with the best results when compared to the results of the other assemblies. All primer pairs mapped to the new assembly and those mapping primer produced products that are almost as the same as what was achieved in wet lab experiments.

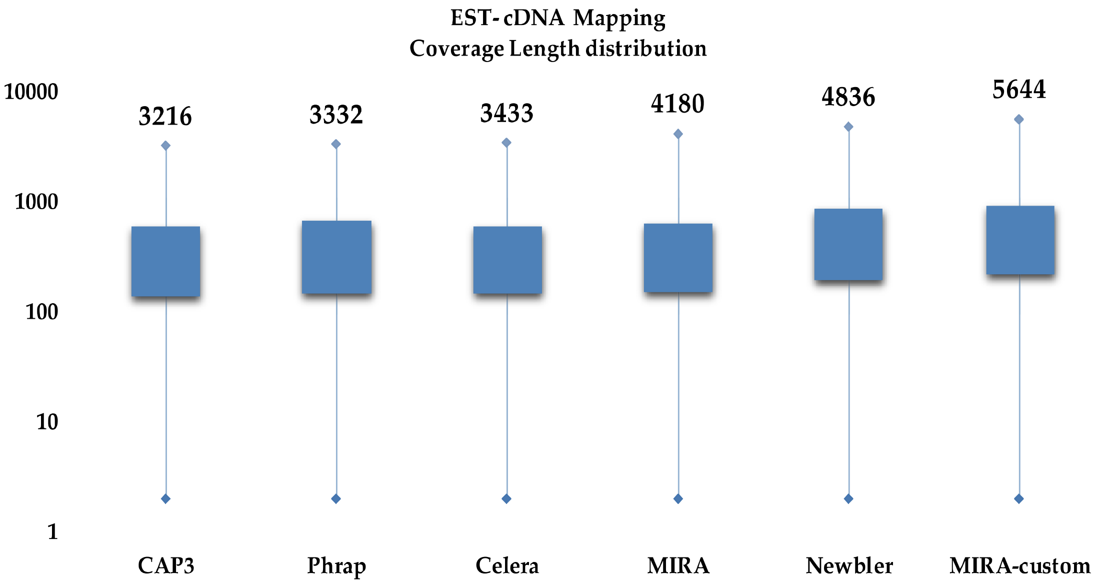

Figure 19. Mapping results of EST- cDNA sequences to all assemblies.

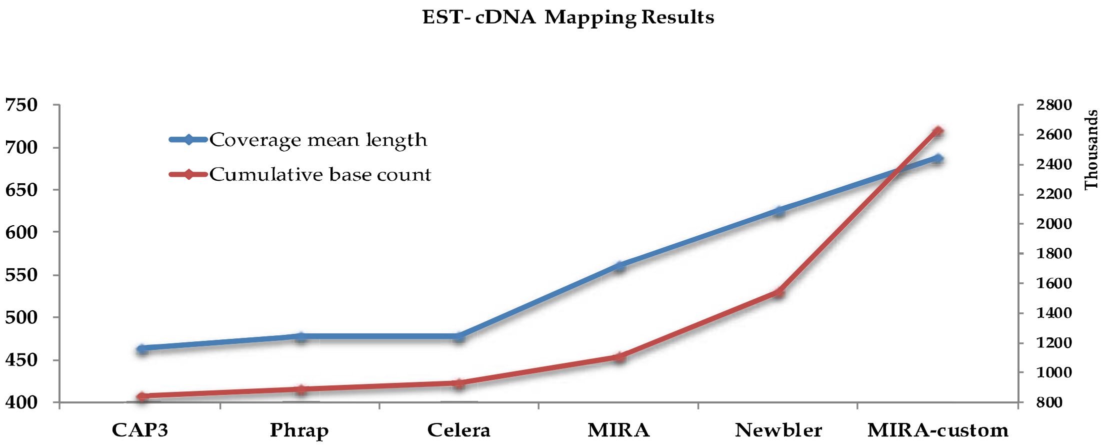

Figure 20. EST- cDNA sequence coverage results of all assemblies.

Mapping EST- cDNA sequences to assemblies analysis revealed that custom assembly settings were accurately defined, based on almost 410% higher cumulative coverage gained with respect to the earlier assembly with **MIRA**, indicating a verypromising coverage ratio.

CHAPTER 4

CONCLUSIONS

Results showed that the very same data may become quiet different in various tools' hand. Since calculation is done in silico, in the course of assembly processes the real outcome must be validated.

One of the obvious conclusions would be that defining an assembly's quality with respect to contig length and count is not correct, and even might be deceiving. For higher accuracy, the load on the assembler should be lightened at the stage of sequencing, by running higher coverage processes, which will drastically give more instructions to the assembler to perform less defective assembly.

Especially instructing for 454 sequencing, it is not feasible to set the threshold for overlap calculation processes higher than 30 bp. Also it is because raw sequences might partially be used, or some sequences might have been fairly set and merged into contigs, contig rebuilding steps should be iterated more than once. This step greatly improves the quality of the assembly, yet costs more calculation time.

CHAPTER 5

OUTLOOK

Through the study we performed over 100 assemblies, for only one of the raw datasets, Papaver. Some of those assemblies were performed with the newer or older versions of the tools with default parameters, while most of them were performed with custom set parameters.

We would like to analyze all those assemblies and have a table with the results of analysis proposed in this study. Mapping processes and especially mapping EST-cDNA sequences to the assemblies individually takes a good amount of time and calculation power. From the logs we have from earlier runs project that the completion of all those analysis will not be feasible in less than a couple of months. Hence we would like to collect the data in time and when ready, we will run a covariance analysis in order to elaborate the results. This analysis will allow us to rate and order the current and proposed metrics together. Having the metrics ordered, a researcher without means and time for these calculations, might have a better understanding with which metrics an assembly should present.

After we finalize the first step, we would like to publish the best assembly of those two organisms as draft genomes, with partially annotations including functional annotations etc.

REFERENCES

Alkan, Can, Saba Sajjadian, and Evan E Eichler. 2011. “Limitations of Next-generation Genome Sequence Assembly.” Nat Methods 8 (1): 61–65. doi:10.1038/nmeth.1527.Limitations.

Baker, Monya. *2012. “De Novo Genome Assembly: What Every Biologist Should Know.”* Nature Methods 9 (4) (January): 333–7. doi:10.1038/nmeth.1935. http://www.ncbi.nlm.nih.gov/pubmed/22453908.

Barbazuk, W Brad, Scott J Emrich, Hsin D Chen, Li Li, and Patrick S Schnable. 2007. *“SNP Discovery via 454 Transcriptome Sequencing.”* The Plant Journal: for Cell and Molecular Biology 51 (5) (September): 910–8. doi:10.1111/j.1365-313X.2007.03193.x. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2169515&tool=pmcentrez&rendertype=abstract.

Bartels, Daniela, Sebastian Kespohl, Stefan Albaum, Tanja Drüke, Alexander Goesmann, Julia Herold, Olaf Kaiser, et al. 2005. “BACCardI--a Tool for the Validation of Genomic Assemblies, Assisting Genome Finishing and Intergenome *Comparison.”* Bioinformatics (Oxford, England) 21 (7) (April 1): 853–9. doi:10.1093/bioinformatics/bti091.http://www.ncbi.nlm.nih.gov/pubmed/15514001.

Bentley, David R. 2006. “Whole-genome Re-*sequencing.”* Current Opinion in Genetics and Development 16 (6) (December): 545–552.

Bocker, S. 2003. “Sequencing from Compomers: Using Mass Spectrometry for DNA De-*novo Sequencing of 200+ Nt.” In* 3rd Workshop on Algorithms in Bioinformatics, ed. Lecture Notes in Computer Science, 476–497. Budapest, Hungary: Springer-Verlag, Berlin.

Cancer, German. 2005. “MIRA: An Automated Genome and EST Assembler.”

Chaisson, Mark, Pavel Pevzner, and Haixu Tang. 2004. “Fragment Assembly with Short Reads.” Bioinformatics (Oxford, England) 20 (13) (September 1): 2067–74. doi:10.1093/bioinformatics/bth205. ttp://www.ncbi.nlm.nih.gov/pubmed/15059830.

Chevreux, B. “Genome Sequence Assembly Using Trace Signals and Additional Sequence Information” (1977).

Chevreux, Bastien. “Sequence Assembly with MIRA3 The Definitive Guide.”

Chevreux, Bastien, Thomas Pfisterer, Bernd Drescher, Albert J Driesel, Werner E G *Müller, Thomas Wetter, and Sándor Suhai. 2004. “Using the miraEST Assembler* for Reliable and Automated mRNA Transcript Assembly and SNP Detection in *Sequenced ESTs.”* Genome Research 14 (6) (June): 1147–59. doi:10.1101/gr.1917404.

Couronne, Olivier, Alexander Poliakov, Nicolas Bray, Tigran Ishkhanov, Dmitriy *Ryaboy, Edward Rubin, Lior Pachter, and Inna Dubchak. 2003a. “Strategies and* Tools for Whole-Genome Alignments Strategies and Tools for Whole-Genome A*lignments”: 73*–80. doi:10.1101/gr.762503.

Dear, Simon, Richard Durbin, Ladeana Hillier, Gabor Marth, Jean Thierry-mieg, and *Richard Mott. 1998. “Sequence Assembly with CAFTOOLS.”* Genome Research: 260–267.

Delcher, a L, S Kasif, R D Fleischmann, J Peterson, O White, and S L Salzberg. 1999. *“Alignment of Whole Genomes.”* Nucleic Acids Research 27 (11): 2369–76.

Delcher, Arthur L, Adam Phillippy, Jane Carlton, and Steven L Salzberg. 2002. “Fast Algorithms for Large-*scale Genome Alignment and Comparison.”* Nucleic Acids Research 30 (11) (June 1): 2478–83.

Earl, Dent, Keith Bradnam, John St John, Aaron Darling, Dawei Lin, Joseph Fass, *Hung On, et al. 2011. “Assemblathon 1* : A Competitive Assessment of De Novo *Short Read Assembly Methods”: 2224*–2241. doi:10.1101/gr.126599.111.Freely.

Enome, W Hole, and D N A*S Equencing. 1999. “B IOLOGY”: 33*–43.

Ewing, Brent, and Phil Green. 1998. “Base-Calling of Automated Sequencer Traces *Using Phred . II . Error Probabilities”: 186*–194. doi:10.1101/gr.8.3.186.

Executive, U K, and Hapmap Ceu. 2007. “Meeting Report : A Workshop to Plan a *Deep Catalog of Human Genetic Variation.”*

Fiers, W, R Contreras, F Duerinck, G Haegeman, D Iserentant, J Merregaert, W Min *Jou, et al. 1976. “Complete Nucleotide Sequence of Bacteriophage MS2 RNA:* Primary and Secondary Structure of the Replica*se Gene.”* \nat 260: 500–507. doi:10.1038/260500a0.

Fleischmann, R D, M D Adams, O White, R a Clayton, E F Kirkness, a R Kerlavage, C *J Bult, J F Tomb, B a Dougherty, and J M Merrick. 1995. “Whole*-genome Random Sequencing and Assembly of Haemophilus Influen*zae Rd.”* Science (New York, N.Y.) 269 (5223) (July 28): 496–512.

Genomics, Celera. 2004a. “Finishing the Euchromatic Sequence of the Human Genome.” Nature 431 (7011) (October 21): 931–45. doi:10.1038/nature03001.

Gordon, D, C Desmarais, and P Green. 2001. *“Automated Finishing with Autofinish.”* Genome Research 11 (4) (April): 614–25. doi:10.1101/gr.171401. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=311035&tool=pmcentrez&rendertype=abstract.

Green, Phil. 1997. “Phrap-.”

Hartl, Daniel L. 1996. *“Sequence Scanning* : A Method for Rapid Sequence Acquisition from Large-*fragment DNA Clones” 93 (February): 1694*–1698.

Huang, X, and a Madan. 1999. “CAP3: A DNA Sequence Assembly Program.” Genome Research 9 (9) (September): 868–77.

Huang, X. 1999. “CAP3: A DNA Sequence Assembly Program.” Genome Research 9 (9) (September 1): 868–877. doi:10.1101/gr.9.9.868.

Huang, Xiaoqiu, Jianmin Wang, Srinivas Aluru, Shiaw-Pyng Yang, and LaDeana *Hillier. 2003. “PCAP: a Whole-genome Assembly Program.”* Genome Research 13 (9) (September): 2164–70. doi:10.1101/gr.1390403.

Huse, Susan M, Julie a Huber, Hilary G Morrison, Mitchell L Sogin, and David Mark *Welch. 2007. “Accuracy and Quality of Massively Parallel DNA Pyrosequencing.”* Genome Biology 8 (7) (January): R143. doi:10.1186/gb-2007-8-7-r143.

Istrail, Sorin, Granger G Sutton, Liliana Florea, Aaron L Halpern, M Clark, Ross *Lippert, Brian Walenz, et al. 2012. “Whole*-genome Shotgun Assembly of Human *Genome Assemblies and Comparison.”* Sciences-New York. doi:10.1073/pnas.0307971.

Istrail, Sorin, Granger G Sutton, Liliana Florea, Aaron L Halpern, Clark M Mobarry, *Ross Lippert, Brian Walenz, et al. 2004. “Whole*-genome Shotgun Assembly and *Comparison of Human Genome Assemblies.”* Proceedings of the National Academy of Sciences of the United States of America 101 (7) (February 17): 1916–21. doi:10.1073/pnas.0307971100.

Jaffe, David B, Jonathan Butler, Sante Gnerre, Evan Mauceli, Kerstin Lindblad-Toh, *Jill P Mesirov, Michael C Zody, and Eric S Lander. 2003a. “Whole*-genome Sequence A*ssembly for Mammalian Genomes: Arachne 2.”* Genome Research 13 (1) (January): 91–6. doi:10.1101/gr.828403.

Jurka, Jerzy, Vladimir V Kapitonov, Oleksiy Kohany, and Michael V Jurka. 2007. *“Repetitive Sequences in Complex Genomes: Structure and Evolution.”* Annual Review of Genomics and Human Genetics 8 (January): 241–59. doi:10.1146/annurev.genom.8.080706.092416.

Kent, W. J. 2002. “BLAT---The BLAST-*Like Alignment Tool.”* Genome Research 12 (4) (March 20): 656–664. doi:10.1101/gr.229202.

Kumar, Sujai, and Mark *L Blaxter. 2010a. “Comparing De Novo Assemblers for 454 Transcriptome Data.”* BMC Genomics 11 (1): 571.

L, Wang, and Jiang T. 1994. “On the Complexity of Multiple Sequence Alignment.” Journal of Computational Biology : a Journal of Computational Molecular Cell Biology 1 (4): 337–348.

Langmead, Ben, Cole Trapnell, Mihai Pop, and Steven L Salzberg. 2009. “Ultrafast and Memory-*efficient Alignment of Short DNA Sequences to the Human Genome.”* Genome Biology 10 (3) (January): R25. doi:10.1186/gb-2009-10-3-r25.

Li, Shuyu, Jiayu Liao, Gene Cutler, Timothy Hoey, John B. Hogenesch, Michael P. *Cooke, Peter G. Schultz, and Xuefeng Bruce Ling. 2002. “Comparative Analysis of* Human Genome Assemblies Reveals Genome-Level Differences.*”* Genomics 80 (2) (August): 138–139. doi:10.1006/geno.2002.6824.

Li, Zhenyu, Yanxiang Chen, Desheng Mu, Jianying Yuan, Yujian Shi, Hao Zhang, Jun *Gan, et al. 2011. “Comparison of the Two Major Classes of Assembly Algorithms* : Overlap ^ Layout ^ Consensus and De-bruijn-*graph” 11 (1).* doi:10.1093/bfgp/elr035.

Lin, Yong, Jian Li, Hui Shen, Lei Zhang, Christopher J Papasian, and Hong-wen Deng. *2011a. “Comparative Studies of De Novo Assembly Tools for Next*-generation Sequencing T*echnologies” 27 (15): 2031*–2037. doi:10.1093/bioinformatics/btr319.

Lin, Yong, Jian Li, Hui Shen, Lei Zhang, Christopher J Papasian, and Hong-Wen Deng. *2011b. “Comparative Studies of De Novo Assembly Tools for Next*-generation *Sequencing Technologies.”* Bioinformatics (Oxford, England) 27 (15) (August 1): 2031–7. doi:10.1093/bioinformatics/btr319.

Liu, Lin, Yinhu Li, Siliang Li, Ni Hu, Yimin He, Ray Pong, Danni Lin, Lihua Lu, and *Maggie Law. 2012. “Comparison of Next-generation Sequencing Systems.”* Journal of Biomedicine & Biotechnology 2012 (January): 251364. doi:10.1155/2012/251364.

Malde, Ketil, Eivind Coward, and Inge Jonassen. 2005. “A Graph Based Algorithm for Generating EST Consensus Sequences.” Bioinformatics (Oxford, England) 21 (8) (April 15): 1371–5. doi:10.1093/bioinformatics/bti184.

Margulies, Marcel, Michael Egholm, William E Altman, Said Attiya, Joel S Bader, Lisa *A Bemben, Jan Berka, et al. 2005. “Genome Sequencing in Microfabricated High-density Picolitre Reactors” 437 (September): 376*–381. doi:10.1038/nature03959.

Materials, S I, Accuprime Taq Dna, High Fidelity, Fosmid Dna, Epicentre Biotechnologies, Plasmid Mega Kit, Solid Phase Re-, The Pcr, Genomic Dna, and *Qiagen Qiafilter. “Supporting Information”: 1*–9.

Maxam, Allan. 1973. “The Nucleotide Sequence of the Lac Operator” 70 (12): 3581– 3584.

Metzker, Michael L. 2010. “Sequencing Technologies — the Next Generation” 11 (jANuARy). doi:10.1038/nrg2626.

Morin, Ryan, Matthew Bainbridge, Anthony Fejes, Martin Hirst, Martin Krzywinski, Trevor Pugh, Helen McDonald, Richard Varhol, Steven Jones, and Marco Marra. *2008. “Profiling the HeLa S3 Transcriptome Using Randomly Primed cDNA and* Massively Parallel Short-*read Sequencing.”* BioTechniques 45 (1) (July): 81–94. doi:10.2144/000112900.

Myers, Eugene W, Granger G Sutton, Art L Delcher, Ian M Dew, Dan P Fasulo, *Michael J Flanigan, Saul A Kravitz, et al. 2012. “Of Drosophila.”*

Narzisi, Giuseppe, and Bud Mishra. 2011. “Comparing De Novo Genome Assembly: The Long and Short of It.” Ed. Stein Aerts. PLoS ONE 6 (4) (April 29): e19175. doi:10.1371/journal.pone.0019175http://dx.plos.org/10.1371/journal.pone.001917.

Ng, P. C. 2003. “SIFT: Predicting Amino Acid Changes That Affect Protein Function.” Nucleic Acids Research 31 (13) (July 1): 3812–3814. doi:10.1093/nar/gkg509. http://www.nar.oupjournals.org/cgi/doi/10.1093/nar/gkg509.

Ng, Pauline C, and Steven Henikoff. 2003. “SIFT: Predicting Amino Acid Changes *That Affect Protein Function” 31 (13): 3812*–3814. doi:10.1093/nar/gkg509.

Notredame, C, D G Higgins, and J Heringa. 2000. “T-Coffee: A Novel Method for Fast *and Accurate Multiple Sequence Alignment.”* Journal of Molecular Biology 302 (1) (September 8): 205–17. doi:10.1006/jmbi.2000.4042.

Palmer, Lance E, Mathaeus Dejori, Randall Bolanos, and Daniel Fasulo. 2010. *“Improving De Novo Sequence Assembly Using Machine Learning and Comparative Genomics for Overlap Correction.”* BMC Bioinformatics 11 (January): 33. doi:10.1186/1471-2105-11-33.

Pareek, C S, R Smoczynski, and A Tretyn. 2011. “Sequencing Technologies and Genome Sequencing.” JOURNAL OF APPLIED GENETICS 52 (4): 413–435. doi:10.1007/s13353-011-0057-x.

Parra, Genis, Keith Bradnam, Zemin Ning, Thomas Keane, and Ian Korf. 2009. *“Assessing the Gene Space in Draft Genomes” 37 (1): 289*–297. doi:10.1093/nar/gkn916.

Paszkiewicz, Konrad, and David J Studholme. 2010. “De Novo Assembly of Short Sequence Reads.” Briefings in Bioinformatics 11 (5) (September): 457–72. doi:10.1093/bib/bbq020. http://www.ncbi.nlm.nih.gov/pubmed/20724458.

Petrov, N a, V a Karginov, N N Mikriukov, O I Serpinski, and V V Kravchenko. 1981. *“Complete Nucleotide Sequence of the Bacteriophage Lambda DNA Region Containing Gene Q and Promoter pR’.”* FEBS Letters 133 (2): 316–20.

Pevzner, P a, H Tang, and M S Waterman. 2001. “An Eulerian Path Approach to DNA Fragment Assembly.” Proceedings of the National Academy of Sciences of the United States of America 98 (17) (August 14): 9748–53. doi:10.1073/pnas.171285098.

Pevzner, Pavel a, Paul a Pevzner, Haixu Tang, and Glenn Tesler. 2004. “De Novo Repeat Classification and Fragment Assembly.” Genome Research 14 (9) (September): 1786–96. doi:10.1101/gr.2395204.

Phase, Proj*ect. 2011. “1000 Genomes Browser Orientation.”*

Pop, Mihai. 2009a. “Genome Assembly Reborn: Recent Computational Challenges.” Briefings in Bioinformatics 10 (4) (July): 354–66. doi:10.1093/bib/bbp026.

Pop, Mihai, Adam Phillippy, Arthur L Delcher, and Steven L Salzberg. 2004. *“Comparative Genome Assembly.”* Briefings in Bioinformatics 5 (3) (September):

Problem, The Biological. “8 DNA Sequence Assembly 8.1.” DNA Sequence.

Quail, Michael a, Miriam Smith, Paul Coupland, Thomas D Otto, Simon R Harris, Thomas R *Connor, Anna Bertoni, Harold P Swerdlow, and Yong Gu. 2012. “A* Tale of Three Next Generation Sequencing Platforms: Comparison of Ion Torrent, *Pacific Biosciences and Illumina MiSeq Sequencers.”* BMC Genomics 13 (1) (January): 341. doi:10.1186/1471-2164-13-341.

Ramos, Rommel Tj, Adriana R Carneiro, Jan Baumbach, Vasco Azevedo, Maria Pc *Schneider, and Artur Silva. 2011. “Analysis of Quality Raw Data of Second Generation Sequencers with Quality Assessment Software.”* BMC Research Notes 4 (1) (January): 130. doi:10.1186/1756-0500-4-130.

Rodrigue, Sébastien, Arne C. Materna, Sonia C. Timberlake, Matthew C. Blackburn, Rex R. Malmstrom, Eric J. Alm, and Sallie W. Chisholm. 2010. “Unlocking Short Read Sequencing for Metagenomics.” Ed. Jack Anthony Gilbert. PLoS ONE 5 (7) (July 28): e11840. doi:10.1371/journal.pone.0011840.

Sahli, Mohammed, and Tetsuo Shibuya. 2012. “Arapan-S: a Fast and Highly Accurate Whole-*genome Assembly Software for Viruses and Small Genomes.”* BMC Research Notes 5 (1) (January): 243. doi:10.1186/1756-0500-5-243.

Salzberg, Steven L, and James a Yorke. 2005. “Beware of Mis-assembled Genomes.” Bioinformatics (Oxford, England) 21 (24) (December 15): 4320–1. doi:10.1093/bioinformatics/bti769.

Sanger F. 1977. “Nucleotide Sequence of Bacteriophage Phi X174 DNA.” Nature 252 (5013): 687–695. doi:doi:10.1038/265687a0.

Sanger, F, and A R Coulson. 1975. “A Rapid Method for Determining Sequences in DNA by Primed Synthesis with DNA Polymerase.” Journal of Molecular Biology 94 (3): 441–446. doi:http://dx.doi.org/10.1016/0022.

Sanger, F, S Nicklen, and A R Coulson. 1977. “DNA Sequencing with Chain-terminating Inhibitors.” Proceedings of the National Academy of Sciences of the United States of America 74 (12): 5463–5467.

Schatz, Michael. 2006. “Celera Assembler Celera Assembler Overview.”

Scheibye-Alsing, K, S Hoffmann, a Frankel, P Jensen, P F Stadler, Y Mang, N *Tommerup, et al. 2009. “Sequence Assembly.”* Computational Biology and Chemistry 33 (2) (April): 121–36. doi:10.1016/j.compbiolchem.2008.11.003.

Schnable, Patrick S, Doreen Ware, Robert S Fulton, Joshua C Stein, Fusheng Wei, *Shiran Pasternak, Chengzhi Liang, et al. 2009. "The B73 Maize Genome: Complexity, Diversity, and Dynamics."* Science (New York, N.Y.) 326 (5956) (November 20): 1112–5. doi:10.1126/science.1178534.

Schuster, Stephan C. 2008a. "Next-generation Sequencing Transforms Today ' s Biology" 5 (1): 16–19. doi:10.1038/NMETH1156.

Shendure, Jay, and Hanlee Ji. 2008a. "Next-generation DNA Sequencing." Nature Biotechnology 26 (10): 1135–1145.

Staden*, R. 1980. "A New Computer Method for the Storage and Manipulation of DNA Gel Reading Data."* Nucleic Acids Research 8 (16) (August 25): 3673–94.

Steven, L. 2002a. "Sequence Assembly : Algorithms and Issues Algorithms That Can Assemble Millions of Small DN*A Fragments into Gene."* Finishing (July): 47–54.

Sugarbaker, David J, William G Richards, Gavin J Gordon, Lingsheng Dong, Assunta *De Rienzo, Gautam Maulik, Jonathan N Glickman, et al. 2008. "Transcriptome* Sequencing of Malignant Pleural Mesothelioma Tumors*."* Proceedings of the National Academy of Sciences of the United States of America 105 (9) (March 4): 3521–6. doi:10.1073/pnas.0712399105.

Sundquist, Andreas, Mostafa Ronaghi, Haixu Tang, Pavel Pevzner, and Serafim *Batzoglou. 2007. "Whole*-genome Sequencing and Assembly with High-throughput, Short-*read Technologies."* PloS One 2 (5) (January): e484. doi:10.1371/journal.pone.0000484.

Tekin, Pelin. 2011. "DETERMINATION OF GENETIC DIVERSITY OF TURKISH SESAMUM (Sesamum Indicum L .) BY USING AFLP MARKERS." IZTECH (June).

Treangen, Todd J., and Steven L. Salzberg. 2011. "Repetitive DNA and Next-generation *Sequencing: Computational Challenges and Solutions."* Nature Reviews Genetics 13 (November 2011) (November 29). doi:10.1038/nrg3117.

Tucker, Tracy, Marco Marra, and Jan M Friedman. 2009. "Massively Parallel Sequencing: The Next Big Thing in Genetic Medicine." American Journal of Human Genetics 85 (2) (August): 142–54. doi:10.1016/j.ajhg.2009.06.022.

Venter, J Craig, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, *Granger G Sutton, Hamilton O Smith, et al. 2001. "The Sequence of the Human Genome" 291 (February).*

W. MIN JOU, G. HAEGEMAN, M. YSEBAERT & W. FIERS. 1972. "Nucleotide Sequence of the Gene Coding for the Bacteriophage *MS2 Coat Protein."* Nature (237): 82–88. doi:doi:10.1038/237082a0.

Walking, Primer. 1998. "Sequence Assembly From ' Encyclopedia of the Human Genome '." Source.

Watson, JD., and FH Crick. 1953. “The Structure of DNA.” Cold Spring Harb. Symp. Quant. Biol 18: 23–31. doi:doi:10.1101/SQB.1953.018.01.020.

Weber, Andreas P M, Katrin L Weber, Kevin Carr, Curtis Wilkerson, and John B *Ohlrogge. 2007. “Sampling the Arabidopsis Transcriptome with Massively Parallel Pyrosequencing.”* Plant Physiology 144 (1) (May): 32–42. doi:10.1104/pp.107.096677.

Weigel, P H, P T Englund, K Murray, and R W Old. 1973. “The 3’-terminal Nucleotide *Sequences of Bacteriophage Lambda DNA.”* Proceedings of the National Academy of Sciences of the United States of America 70 (4) (April): 1151–5.

Wheeler, David a, Maithreyan Srinivasan, Michael Egholm, Yufeng Shen, Lei Chen, *Amy McGuire, Wen He, et al. 2008. “The Complete Genome of an Individual by Massively Parallel DNA Sequencing.”* Nature 452 (7189) (April 17): 872–6.

Wu, Xue, Woei-jyh Adam Lee, Damayanti Gupta, and Chau-*wen Tseng. “ESTmapper* : *Efficiently Clustering EST Sequences Using Genome Maps”: 1*–12.

Zerbino, Daniel R, and Ewan Birney. 2008a. “Velvet: Algorithms for De Novo Short Read Assembly Using De Bruijn Graphs.” Genome Research 18 (5) (May): 821–9. doi:10.1101/gr.074492.107.

Zhang, Wenyu, Jiajia Chen, Yang Yang, Yifei Tang, Jing Shang, and Bairong Shen. *2011. “A Practical Comparison of De Novo Genome Assembly Software Tools for* Next-*generation Sequencing Technologies.”* PloS One 6 (3) (January): e17915. doi:10.1371/journal.pone.0017915.

Zhang, Z, S Schwartz, L Wagner, and W Miller. 2000. “A Greedy Algorithm for Aligning DNA Sequences.” Journal of Computational Biology : a Journal of Computational Molecular Cell Biology 7 (1-2): 203–14. doi:10.1089/10665270050081478.

Çelik, İbrahim. 2011. “DEVELOPMENT OF SSR MARKERS IN POPPY (Papaver Somniferum L .) in Molecular Biology and Genetics.” IZTECH (June).