Draft assembly of the genome of the emergent orchid model *Erycina pusilla*

Genomics & Sequencing

Year 3, September 2014 – February 2015, 1st and 2nd term, 8 EC

Authors: Dennis Middendorp, Josje Buren and Jan Willem Wijnands

Supervisors: Dr. Barbara Gravendeel, Dr. Rutger Vos and Anita Dirks   
(Naturalis Biodiversity Center)

Index

[Abstract 4](#_Toc410299633)

[Introduction 5](#_Toc410299634)

[The emergent orchid model *Erycina pusilla* 5](#_Toc410299635)

[PacBio 5](#_Toc410299636)

[Illumina 5](#_Toc410299637)

[Mapping Illumina against PacBio reads 6](#_Toc410299638)

[Goal 6](#_Toc410299639)

[Research question 6](#_Toc410299640)

[Materials and Methods 7](#_Toc410299641)

[Literature study 7](#_Toc410299642)

[Different datasets used 7](#_Toc410299643)

[Puppet 7](#_Toc410299644)

[OpenStack 7](#_Toc410299645)

[About Burrows-Wheeler Aligner 8](#_Toc410299646)

[About CLC Genomics Workbench 8](#_Toc410299647)

[About SAMtools 8](#_Toc410299648)

[About the MixedReadsAssembyErycinaBWA-pipeline 8](#_Toc410299649)

[BLAST 9](#_Toc410299650)

[Fasta1line.pl 9](#_Toc410299651)

[Results 10](#_Toc410299652)

[Literature study 10](#_Toc410299653)

[Work environment design 11](#_Toc410299654)

[Puppet manifest 11](#_Toc410299655)

[Puppet installation 11](#_Toc410299656)

[OpenStack 12](#_Toc410299657)

[Assembly script 12](#_Toc410299658)

[Assembly 12](#_Toc410299659)

[BLAST 14](#_Toc410299660)

[Discussion 16](#_Toc410299661)

[Work environment 16](#_Toc410299662)

[Assembly script 16](#_Toc410299663)

[Local BLAST search for PISTILLATA promoter region 17](#_Toc410299664)

[Conclusion 18](#_Toc410299665)

[Acknowledgements 18](#_Toc410299666)

[References 19](#_Toc410299667)

[Appendices 21](#_Toc410299668)

[List of Figures 21](#_Toc410299669)

[Appendix A: Puppet manifest 25](#_Toc410299670)

[Appendix B: Puppet installation 28](#_Toc410299671)

[Appendix C: Assembly Script 29](#_Toc410299672)

[Appendix D: Fasta1line.pl 32](#_Toc410299673)

[Appendix E: PISTILLATA known gene sequence used as reference 33](#_Toc410299674)

[Appendix F: blast result of 16 day run consensus 34](#_Toc410299675)

[Appendix G: blast result of 10 longest PacBio contigs consensus 35](#_Toc410299676)

[Appendix H: blast result of one of the longest PacBio contigs consensus 36](#_Toc410299677)

[Appendix I: blast result all PacBio contigs (Layout gefixed in word) 37](#_Toc410299678)

[Appendix J: Overview of different datasets 45](#_Toc410299679)

# Abstract

Background: This project was carried out to build a work environment for creating a first draft assembly of the genome of the emergent orchid model *Erycina pusilla* from different Next Generation sequence data and use this assembly to find more of the promoter region of the PISTILLATA gene than currently known. The reason this research project focused on the genome of *Erycina pusilla* is because this species is the topic of the PhD project of Anita Dirks focusing on functional genomics of orchids. The reason that we had to look for larger parts of the PISTILLATA gene is because Anita Dirks wants to use the promoter sequence to create a knockout of the PISTILLATA gene in order to prove that it codes for flower shape.

Methods: Firstly, we built a work environment which is hosted on the OpenStack cloud server of Naturalis Biodiversity Center. This work environment consisted of an instance and an additional hard drive to store data. The NGS data that we assembled during this project consisted of one PacBio dataset and three Illumina datasets. For the draft assembly, we mapped the relatively short Illumina reads against the ten largest PacBio contigs. In order to do this we used a script that was created for a past project. We modified this script to work with our data. This script was then used to create a draft assembly in the form of a FASTA consensus sequence. A separate BLAST command was used to search for the PISTILLATA gene in this consensus sequence, each of the ten longest PacBio contigs, their consensus, and all of the PacBio data.

Results: It took 16 days to create a consensus that contained only the Illumina reads that mapped against all PacBio contigs in the order of the contigs analysed, i.e. from short too long. There were also no matches retrieved with the PISTILLATA sequence blasted against the consensus sequence of the Illumina reads mapped against the ten longest PacBio contigs. We did get some good hits when we blasted the PISTILLATA sequence against only the PacBio data though.

Discussion: It has become very clear to us that our both our work environment and assembly script still need improvement. Even though we did not succeed in our initial goals of making a draft assembly to find more of the promoter region of the PISTILLATA gene we are of the opinion that we have succeeded in laying a firm foundation for future analyses of the genome data generated of *Erycina pusilla*.

Keywords: MADS box genes, Next Generation sequencing, OpenStack work environment

# Introduction

With the Human Genome project[1], science took a huge step forward in the field of genomics. Never before had a genome of this size been ‘completely’ mapped. Since then, science has advanced and created new methods so that entire genomes can be mapped faster and cheaper. But even today mapping an entire genome is no small task, especially when no reference genome is available. Researchers can spend years on a project focusing on just one genome. One particular genome that is currently being investigated is the one belonging to the emerging orchid model *Erycina pusilla*. This project was carried out to build a work environment for creating a first draft assembly of the genome of the emergent orchid model *Erycina pusilla* and use this assembly to find more of the promoter region of the PISTILLATA gene than currently known. The reasons for doing this are outlined below in more detail.

## The emergent orchid model *Erycina pusilla*

*Erycina* is an orchid genus and closely related to the genus *Oncidium* that belongs to the tribe Cymbidieae, subtribe Oncidiinae within subfamily Epidendroideae. *Erycina* is different in terms of morphology and physiology from, for instance, *Phalaenopsis* orchids of which the genome is also being investigated. The genome of this species is diploid and contains just six chromosomes. Anita Dirks[2] chose to do genomics of *Erycina pusilla* because of its small, relatively simple and fully sequenced genome. *Erycina pusilla* can be grown fast and will produce flowers and fruits *in vitro* in just one year[3]. These aspects make *Erycina pusilla* not only an attractive model plant for functional genomic and flowering studies of *Oncidium*, a related genus of great economic significance, but also an excellent parent for traditional hybridization methods.[4]

For her PhD research, Anita Dirks is looking for the promoter regions of MADS box genes in the genome of *Erycina pusilla*. These genes are responsible for the development of floral organs. Anita wants to find proof for the function of these genes by using a knockout of the individual genes and investigating the resulting phenotypes. For this method, the full promoter region needs to be uncovered. The development of Next Generation Sequencing (NGS) now makes this possible. Below, we will explain the particular methods used for this in more detail.

## PacBio

Pacific Biosciences[5] was founded in 2004 and develops systems that can be used for gene sequencing. One of their machines, the “PacBio RS II”, was used to sequence reads of *Erycina pusilla*, which have been compounded into contigs, ranging in size from 51 bp to 40616 bp. This machine uses single molecule real time sequencing (SMRT), a visual explanation of this process is found in figure 1 in the appendix. SMRT uses a SMRT chip that contains a lot of zero-mode waveguides (ZMW). These ZMW are circular holes in an aluminium film. Inside each of these ZMW a single DNA molecule, together with a DNA polymerase, is immobilized to the bottom of the hole. This allows for the monitoring of the signal given by the incorporation of phospho-linked nucleotides in real time.[6]

## Illumina

Illumina[7] was founded in 1998 and is a developer of systems that are used for genetic and biological analysis. One of their machines, the “HiSeq X Ten”, has been used to sequence reads of *Erycina pusilla*. The Illumina systems work by making use of the four bases which have all been fluorescently labelled with a different colour. The visual explanation of this process is found in figure 2 in the appendix.

The systems use the following steps:[8]

1. Add all fluorescently labelled bases to the DNA.
2. Bases will incorporate where they are needed.
3. Non-incorporated bases are washed away.
4. A laser excites the dyes and a photo is taken of the bases.
5. The dyes are removed from the incorporated bases
6. Repeat until fully sequenced.

An Illumina HiSeq was used to generate reads of *Erycina pusilla*, ranging in size from 100 bp to 101 bp. It should be noted that the reads that have a length of 101 bp start with an extra ‘N’.

## Mapping Illumina against PacBio reads

For a first draft assembly of the genome of *Erycina pusilla*, contigs will need to be created from the Illumina reads and PacBio contigs using reference mapping. This is a big undertaking that may take many weeks and/or months to complete as the size of the data to be processed is quite large. In order to lay the foundation for this project, we created a work environment and a script for draft assembly of the contigs and local blasts to look for sequences of the MADS box PISTILLATA gene so others may use it for further assembly and analysis of the genome of *Erycina pusilla*.

# Goal

The main goal of this project was to create an OpenStack work environment that other researchers may use for further genomic research on *Erycina pusilla*. All data generated from the genome of *Erycina pusilla* by BGI and LGTC (Illumina) and PRI WUR (PacBio), in addition to the Orchidstra database (Illumina Expressed Sequence Tags) generated in Taiwan are now available in this work environment. The most common assembly tools and other commonly used bio-informatics tools for genome research such as BLAST were installed on this work environment. Working scripts are available to generate assemblies from Illumina reads and PacBio contigs and search for sequences of the promoter region of the MADS box B class PISTILLATA[9] gene in the resulting consensus.

# Research question

This project focused on finding an answer to the following research question: can we retrieve parts of the promoter region of the MADS box B class gene PISTILLATA in a consensus created from all Illumina reads and the ten largest PacBio contigs with our work environment that were previously uncovered?

The hypothesis was that we would retrieve these parts as currently only small parts of the PISTILLATA gene are known. These were all generated using first generation Sanger sequencing. We now analysed Next Generation sequences created from the entire genome so the chances of obtaining previously not yet discovered parts of the PISTILLATA gene were larger.

# Materials and Methods

In order to bring this project to a good conclusion we have employed several methods. All of these are explained in more detail below.

## Literature study

First, we read several articles about the background of this project. We learned how to make an efficient directory layout[10] and how an assembly works[1]. We also read something about the kind of genes we were looking for. These genes are called MADS box B class genes. We specifically looked at the PISTILLATA gene. This gene is expressed in petals and stamens of flowers, which are organs formed in the first and second floral whorl[11]. After that, we read an article about creating assemblies of PacBio and Illumina data[11].

## Different datasets used

In [appendix J](#_Appendix_J:_Overview), an overview is given of the different datasets that were used during this project. These consisted of PacBio and Illumina data generated from *Erycina pusilla*. We also used the Orchidstra[12,13] database in search for (a larger part of) the PISTILLATA gene. The combined data amount to almost 90 GB.

## Puppet

Puppet is a configuration management system that allows users to define a system and then automatically enforces the correct state[14]. With this software, we created a script that automatically installed packages on a physical or virtual machine. This saved a lot of time and prevented errors usually associated with manual installations. The puppet-script installed several standard packages and compiled the latest source code of the Burrows-Wheeler Aligner.

## OpenStack

For this project we created a work environment on a cloud server. The cloud server was hosted with OpenStack software. Naturalis provided us with an OpenStack flavour (basic server template). The fact that we used a cloud server meant that not all software was available to us. The software available is summarized in table 1 in the results. To employ the virtual machine we had to use software that could be run on a virtual machine and from the command line. Available software that met those requirements at the time of this project consisted of Burrows-Wheeler Aligner, SOAPdenovo, Velvet and ABySS.

Besides the virtual machine, Naturalis also assigned us with a working volume of 1000GB. On this volume, we stored our initial data and all data derived from our analyses.

To login on the OpenStack work environment the user had to allocate a security key pair. This file was called a PEM-file and could be downloaded from the OpenStack web-interface. After downloading, the file was stored in the personal home-folder. After that, the user had to modify the permissions of the PEM-file. This was possible with the following command:

|  |
| --- |
| chmod 600 <path/to/file.pem> |

The user could subsequently make an instance. This was the actual work environment. When you pushed the button ‘Launch instance’ you could specify the specifications of the instance. In the first tab, as seen in Figure 3 in the appendix, you had to choose the ‘Availability Zone’. At Naturalis, the standard availability zone was called: ‘Nova’. We have not adapted this setting. After selecting this availability zone, the user could choose between several ‘flavours’. These flavours were defined by Naturalis. Our flavour made use of 64GB of RAM, 8 VCPUs and 160GB of data storage. Then the user needed to choose an image boot source. We chose to boot from image and the newest Ubuntu 14.04.

At the second tab of the screen, the user could define the key pair and import the PEM-file that was made earlier. The admin password could be left blank.

At the third tab of the screen, the user could choose the network that was available. We had no choice as there was just one network available for this instance. This network was called: private\_erycina\_pusilla.

The last tab was the ‘Post-Creation’-tab. Here the user could load its own initialization script (appendix B). This is a shell script, which configures the instance and then calls our puppet script. This puppet script provides commonly used packages and compiled software needed for our research. The puppet-script that we used is included in appendix A.

After all this, the user could press ‘Launch instance’.

After launching the instance and waiting a couple of minutes for OpenStack to be set up, the user had to assign an available floating-IP. The IP-address behind the ‘@’ was called the floating-IP. On the website from OpenStack Naturalis you could assign an available floating-IP. The standard username for login to the virtual machine was: Ubuntu. In order to use the PEM-file you needed to use the argument ‘-i’, which stands for the identifier, in combination with the name of the PEM-file. This allowed you to identify yourself against your virtual machine. Now the user could login to the virtual machine through a terminal-window:

|  |
| --- |
| ssh -i janwillem.pem ubuntu@10.42.1.140 |

After you logged in to the user account on the virtual machine information about the specifications was received. An example is showed in figure 4 included in the appendix. From there the user could execute several commands or programs.

## About Burrows-Wheeler Aligner

BWA is a software package for mapping low-divergent sequences against a large reference genome. It consists of three algorithms: BWA-backtrack, BWA-SW and BWA-MEM. The first algorithm is designed for Illumina sequence reads up to 100bp, while the other two are designed for analysis of longer sequences ranging from 70bp to 1Mbp.[15] We used this application for creating an assembly of Illumina reads and PacBio contigs.

## About CLC Genomics Workbench

This software package is extremely easy to use but only commercially available. The functionalities of the main workbench are used by multiple researchers all over the world for DNA, RNA, and protein sequence data analysis, such as gene expression analysis, primer design, molecular cloning, phylogenetic analyses, and sequence data management. These functionalities are all wrapped in an organized graphical user-interface. We assigned this application as backup plan for perform a local blast but did not use it in the end.

## About SAMtools

SAMtools stands for Sequence Alignment/Map. We used this application to index our PacBio-reference genome. We also used SAMtools for manipulating our SAM-files and merging our BAM-files. SAMtools also have another feature: BCFtools. We used BCFtools to make a VCF-file. From this file we created a consensus of the combined Illumina and PacBio data.

## About the MixedReadsAssembyErycinaBWA-pipeline

Besides designing a work environment from scratch, we designed a pipeline that was able to provide a consensus of the genome of *Erycina pusilla* created by an assembly of 4 datasets provided by 3 companies. The PacBio data came from PRI WUR, one Illumina dataset came from LGTC and the remaining two were produced by BGI. Our pipeline was written in the bash-language. We recycled some codes from another project done in the past. That project produced an assembly of the tomato genome. BWA has a dominant role in our script.

## BLAST

For searching in the consensus we used local blast. This was possible with the package ncbi-blast+. First, we had to use the command ‘makeblastdb’ for creating a local database. An example for the command that we used for creating this database is:

|  |
| --- |
| makeblastdb –in erycina\_consensus.fasta –parse\_seqids –dbtype nucl |

With ‘blastn’ we searched for the PISTILLATA-gene (EPTC010697) in the consensus. An example of the ‘blastn’-command is:

|  |
| --- |
| blastn –db erycina\_consensus.fasta –query EPTC010697.fasta –out results.out |

## Fasta1line.pl

For extracting the 10 longest PacBio-contigs out of the entire PacBio fasta-file, we used two existing scripts available on the internet. First, we used a perl-script (Appendix D)[16] to place all headers and sequences independently from each other in separate lines. The whole script with explanation is included in the appendix.

Secondly, we had to use an example of the following command:

|  |
| --- |
| cat sequences.oneline.fasta | perl -e 'while (<>) {$h=$\_; $s=<>; $seqs{$h}=$s;} foreach $header (reverse sort {length($seqs{$a}) <=> length($seqs{$b})} keys %seqs) {print $header.$seqs{$header}}' > sequences.oneline.biggest1st.fasta |

This command ensured that every header and sequence was sorted from the longest to the shortest. Thereafter we used the ‘head’-command to extract the 10 longest PacBio-contigs.

# Results

Several results were generated during this project, ranging from the creation of a work environment to a script capable of making an assembly to searching for candidate genes in the resulting assembly and individual contigs. The individual results obtained are described in more detail below.

## Literature study

The literature study provided us with the following insights. In table 1 an overview is provided of the different tools found in the literature to combine Illumina with PacBio reads. We chose to work with BWA because it is possible to produce an alignment with several cores when using this tool. Besides that, we chose BWA because it is suited for long reads.

|  |  |  |  |
| --- | --- | --- | --- |
| **Software name** | **Strengths** | **Weaknesses** | **Considerations** |
| SOAPdenovo[17] | Runs on multiple cores | Increased error rate at the end of long reads. | This tool could not be used because it only works with short reads. |
| Velvet[18] | Runs on multiple cores  Fast | Hard to use for new users. | Velvet uses too much RAM memory. Due this reason the tool could not be used. |
| ABySS[19] | Runs on multiple cores  Suited for large genomes | Can’t be used with short reads. | This tool could not be used because it only works with short reads. |
| CLC Workbench Genomics[20] | User-friendly | Expensive, no license for Linux. | CLC was not used because while it is available for Linux it is not runnable from the command line. |
| Burrows-Wheeler Aligner[15] | Long reads versus short reads aligner | This tool is less specific and increased error rate. | This tool was chosen to work with because out of all options, BWA seemed to be the least error-prone. |
| Celera[21] | Long reads versus short reads aligner | Many errors because of missing prerequisites | In our opinion this tool has too much chance to generate errors in the results and therefor hasn’t been used. |

**Table 1: Overview of different software tools available with their strengths and weaknesses.**

## Work environment design

In figure 5 you can see the design of the work environment that we created. This work environment design gave everyone a proper overview of the project.

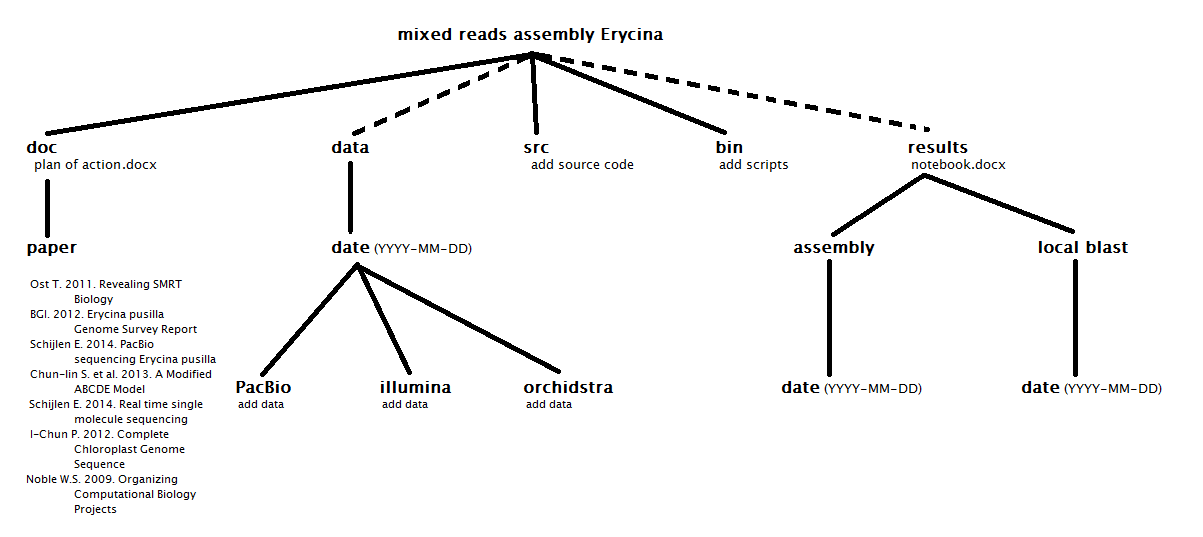


Figure 5: Overview of the created work environment.

The work environment is separated into multiple folders. The doc folder was used to store all kinds of documents pertinent to the project. The data folder was used to store the initial data investigated. The src folder was designed to store third-party code or software. The bin folder was used to store code that was written by members of the project and the results folder was made to store all results that were gathered during the project. The reason why the data and results folder are connected with a broken line is to show that those folders are located on the volume side of the work environment.

This design allowed us to keep track of not only our progress in building the work environment but also the results that came out of using the environment. The puppet manifest enabled us to make sure that the work environment functioned as planned.

## Puppet manifest

The puppet manifest, as shown in appendix A, worked as follows.

Section A defined the different paths used in the manifest. Section B installed the latest version of the different packages used. In section C the default paths for the data, the scripts and the source code were set. Section D created all directories. Section E installed BWA.

To run the manifest, puppet had to be installed in the initialisation script of the instance.

## Puppet installation

To make sure everything worked in the instance, an initialisation script was provided during the creation of the instance. The initialisation script can be found in appendix B.

Section A of the script showed the commands. Then it cloned the manifest. With section B the script installed puppet or updates if needed. Section C moved puppet to its own directory. The last part of the script copied data and called upon the manifest.

With this script the work environment was created in OpenStack.

## OpenStack

We made a work environment to improve the efficiency of the project. The environment was divided in two parts, an instance and a volume. The specifications of these parts are summarised in table 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **RAM** | **VCPUs** | **Storage** | **OS** | **Used for** |
| instance | 64GB | 8 | 160GB | Ubuntu 14.04 | virtual machine |
| Volume | NA | NA | 1000GB | NA | storage |

**Table 2: Overview work environment specifications.**

## Assembly script

A single script pipeline was used to create a consensus of Illumina reads and PacBio contigs generated from the genome of *Erycina pusilla*. In appendix C, the full script is provided. It is divided in different sections that are described in more detail below.

In section A, the reference sequences were defined, next to the reads to be mapped against the reference, the consensus file and the specific samples used for the assembly. It also defined the number of cores used by BWA.

In section B, the reference was indexed, which means that the reference data were pre-processed. This allowed for faster mapping of the samples.

In section C, the loop for the assembly started. The loop was repeated for all the samples in the sample directory. It defined both paired end fastq files and the local variables of the files produced over the run of the script.

In section D, a local Fasta file was made of the reference and the fastq files. It also created the variables for the output file and a SAM file.

In section E, a SAM file was created from the alignment created in section D.

In section F, this SAM file was filtered based on the quality of the mapped reads. The quality threshold that we used was 50. The script also removed unmapped reads.

In section G, a BAM file was generated and indexed from the SAM file generated in section F.

In section H, all created BAM files were merged and the resulting BAM file was indexed.

In section I, a consensus FASTA file and FASTQ file were created using the SAMtools mpileup command. The consensus can contain both upper- and lowercase letters. This happens because bases in lowercase are indications of repeats, indels or insufficient/excessive read depth[23]. It is a process called DNA masking[24]. This is done because if, for example, you want to align multiple DNA sequences with a reference genome you do not want your DNA sequences to match with multiple positions in the reference due to repeats.

## Assembly

The assembly script was run twice and the specifics of these runs are provided in table 3. The full results of the BLAST runs are provided in appendix F to I. The 10 largest PacBio contigs were found by running the Perl script shown in appendix D, and selecting the first 20 lines, which included both the headers and contigs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Run (year- day-month)** | **Data** | **Number of contigs** | **Length of contigs (bp)** | **Number of mapped Illumina reads** | **Duration of the run** |
| 1 (14-12-9) | All Illumina data on all PacBio data | 11682 | not known | not known | 16 days |
| 2 (15-01-16) | All Illumina on 10 largest PacBio contigs | 1 | 8200 | 2 | ~1 day |

**Table 3: Overview of performed assembly runs.**

## BLAST

A total of four blast runs were carried out. The PISTILATA gene, of which the known sequence used as template can be found in appendix E, was used with different data sets as reference. In table 4, more details of these runs are provided.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Run** | **Database** | **# nucleotides in database** | **# sequences in database** | **Query Sequence length (bp)** | **Hits** | **Effective search space (no unit)** |
| 1 (15-01-08) | consensus\_erycina.fasta  (assembly Run 1) | 50,089,952 | 11,682 | 887 | 0 | 42925791524 |
| 2 (15-01-??) | 10 largest PacBio contigs consensus | 25,523 | 1 | 887 | 0 | 22216597 |
| 3 (15-01-??) | 10 largest PacBio contigs separated | 40,615 (of longest contig) | 1 | 887 | 0 (for all contigs) | 35320260 (of longest contig) |
| 4 (15-01-16) | opt\_smrtanalysis\_current\_common\_jobs\_016\_016437\_data\_filtered\_subreads.fasta | 6,639,041,197 | 1,467,2864 | 887 | 14 | 5644083363336 |

**Table 4: Overview of performed BLAST searches.**

In the consensus of all Illumina reads and the ten largest PacBio contigs no hits were found. Only when the reference sequence was blasted against all of the PacBio data, hits were found. These hits were found in ten PacBio contigs (Appendix I), with hits ranging in size between 33 and 399 bp. The following hits were especially interesting because they started within the first 10 bases of the known sequence of the PISTILATA gene, which are close to the promoter region of interest to Anita Dirks:

1. >lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/8830\_11833
2. >lcl|m140504\_121849\_42179\_c100649792550000001823116010071454\_s1\_p0/21360/1347\_14134
3. >lcl|m140501\_161148\_42179\_c100643742550000001823115909101441\_s1\_p0/122586/0\_4550
4. >lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/27127\_29777

An overview of the positions of the mapped PISTILATA gene to these PacBio contigs is provided in table 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Contig** | **1** | **2** | **3** | **4** |
| **Start position PISTILLATA** | 1 | 4 | 4 | 6 |
| **End position PISTILLATA** | 230 | 230 | 230 | 230 |
| **Start position contig** | 2016 | 2542 | 296 | 1960 |
| **End position contig** | 2246 | 2776 | 55 | 2168 |
| **Identities** | 217(92%) | 216(90%) | 215(87%) | 190(83%) |
| **Gaps** | 13(5%) | 16(7%) | 23(9%) | 22(10%) |
| **Length of mapped sequence (bp)** | 237 | 239 | 246 | 228 |
| **Total length of contig (bp)** | 3003 | 12787 | 4550 | 2650 |

**Table 5: BLAST information of interesting contigs.**

# Discussion

All results obtained, ranging from (i) the creation of a work environment to (ii) a script capable of making an assembly of Illumina reads with PacBio contigs to (iii) searching for candidate genes in the resulting assembly and individual contigs, are discussed in more detail below.

## Work environment

The work environment that we created is fully functional but it still requires a few modifications that we could not make ourselves due to lack of time. These are outlined below.

First of all, we created a folder within the work environment containing only symbolic links. These symbolic links refer to all the PacBio and Illumina data that were available to us during this project. Some of these symbolic links need to be deleted because during the process of making the symbolic links some additional links were created that refer to simple readme files that ended up in the same folder as the reference PacBio and Illumina data. This is undesirable as they most likely will not be used and will therefore make it harder to see in an instance to what kind of data the symbolic links refer.

Secondly, the assembly script that we wrote made use of data located in the AllRawData/Samples folder. The location of this Samples folder should be changed in order to keep a more logical layout of the work environment as this AllRawData folder should be removed because this folder currently doubles the amount of space all the data takes up and has become obsolete due to the incorporation of previously mentioned symbolic links.

Thirdly, the work environment was created on the OpenStack instance by connecting with the project’s GitHub depository and downloading the files and folders that were located there. In order for other researchers to get a better understanding of this project the README.md file on GitHub needs to be updated. While it does currently contains some information it needs to be expanded because this will make it easier for future researchers who will continue this project or use it as a reference to understand what this project is or was about.

Lastly, the work environment made use of volumes (hard-drives) for saving data. At the time of this project there was no manual available on how to make a new volume and connect this with the instance. This was not handy because we continuously feared making a volume unusable. Most likely, this project will be continued using OpenStack. We therefore think it will be useful to create a manual for future projects that will make use of OpenStack.

## Assembly script

In order to assemble the *Erycina pusilla* genome from individual Illumina reads and PacBio contigs we created a script. With this script we have generated some first assemblies. Unfortunately, none of these contained any useful information yet for the PhD research of Anita Dirks. The results obtained helped us to define further improvements of our script though. We especially recommend the changes outlined below in more detail.

Firstly, the location of where the results are saved needs to be changed. First, the results were saved on the instance but that gave problems with the amount of memory space used. Currently, they are saved on the volume in the Results folder with the results of every run of the assembly script separated into a different folder that is defined by the time the script was initialized. We recommend using not only the time but also the date, resulting in the following layout: Results/Assembly/Date/Time. An example of this layout is Results/Assembly/2015-01-31/12:34:56. Using this layout ensures that the results of every run are separated and saved in a chronological order with a minimum risk of erasing results of previous runs. Alternatively, epoch time can be used instead of the 12:34:56 time notation because this allows for numerical sorting and because the “:” character can cause problems on some operating systems.

Secondly, in the assembly script there are a couple of steps that pipe the output of one command to the next. This causes the results of these commands to be saved in the same folder as where the input files are located, which is undesirable because of loss of overview of both data and results. The assembly script will need to be changed to correct for this error. The commands that pipe the results to the next command should be split. In this way you can specify the location of the results for every command separately. The other solution would be to use a couple of move commands at the end of the script to move the results to the appropriate save location, as discussed above. We recommend using solution number two because it is the least difficult and has less chance of breaking up the assembly script into too many individual parts. Solution one can always be implemented later on if future researchers think this is necessary.

Thirdly, the assembly script currently generates a consensus sequence containing only the mapped Illumina reads. This is not what we intended. The cause of this is the following command:

samtools view -bS -F 4 -q 20 $SAM > $SAM.filtered

In this command, the argument -F 4 passed to the samtools view command tells the command to remove all the unmapped reads and keep only the mapped reads[22]. What we want is a consensus made of all PacBio contigs next to the mapped Illumina reads. Alternatively, the argument -f 4 could be used. Unfortunately, this does the opposite of -F 4, it keeps the unmapped reads and removes the mapped reads[22]. This is also not what we want. We are not entirely sure how these arguments work but it seems they check for the bits of the integer you pass to the argument in the SAM file[22]. One thing that can be tested is what happens when you pass a different integer to the argument. This can be tested with both arguments. Another thing that might be tested is getting rid of the -F argument all together. We realize this might prompt the command to keep all data (and just filter on mapping quality). While this is still not what we want it would be a step closer to the solution desired.

Fourthly, we have noticed that the filtered SAM files that were generated were relatively small as compared to the original SAM file. This is caused by the removal of the unmapped reads that we spoke of above. In addition to removing the unmapped reads, the current script also filters the remaining mapped reads based on a quality score. At first, we told the command to remove all mapped reads with a quality score of less than 50. This sometimes resulted in empty filtered SAM files. That is why we turned the threshold value down to 20. We suggest further experimenting with this value, i.e. turning it down to 10, after the problem with removing unmapped reads has been solved.

Fifthly, when looking at the consensuses that have been created during this project, we realized that the mapped reads were displayed as a mix of small and capital letters, e.g. atCTtaggtaTTT. This happens for the following reasons: bases in lowercase are indications of repeats, indels or insufficient/excessive read depth[23]. This is done by the samtools mpileup command. It is a process called DNA masking[24]. This is done because if, for example, you want to align multiple DNA sequences with a reference genome you do not want your DNA sequences to match with multiple positions in the reference due to repeats.

There is an additional problem with the generated consensuses. Instead of generating one long consensus sequence, the consensuses are split up into the different PacBio contigs used as references, in the order in which they were offered. So far, we have not been able to discover what causes this so this will need further investigation as well.

Finally, we think that it would be a good idea to use different aligners instead of only BWA in the follow-up research. We have been unable to retrieve statistics such as coverage depth using BWA but other aligners are capable of producing such data. Statistics are needed to assess which data generated are the most reliable and which are less reliable.

## Local BLAST search for PISTILLATA promoter region

The blast results of the PISTILLATA gene against the ten longest PacBio contigs came up empty but when we blasted the same reference sequence against all PacBio contigs we got a couple of (partial) hits. We therefore recommend using the contigs that had matches with the PISTILLATA gene (in addition with maybe a couple of other contigs) for future testing of updated versions of our assembly script. We also highly recommend the use of contig >lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/8830\_11833 in order to try and find more of the promoter region of the PISTILLATA gene. We recommend this specific contig because the starting position of the known PISTILLATA gene maps to position 2016 of the PacBio contig. This means there are about 2000 bases that precede the beginning of the known PISTILLATA sequence and thus offer a good chance to retrieve more of the promoter region.

# Conclusion

Our research question was: Can we retrieve parts of the promoter region of the MADS box B class gene PISTILLATA in a consensus created from Illumina reads and PacBio contigs of the genome of the emergent orchid model *Erycina pusilla* with our work environment that were previously uncovered by Sanger sequencing?

The answer to this question is: not yet because we did not have sufficient time to fine-tune our assembly script. It now only depicts Illumina reads mapping against the PacBio contigs. If this could be adjusted into also depicting the PacBio contigs, additional parts of the MADS box B class gene PISTILLATA not yet found by Sanger sequencing will definitely be retrieved. We know this because we already found a huge part of the PISTILLATA gene in one of the PacBio contigs.

# Acknowledgements

Anita Dirks-Mulder (Naturalis Biodiversity Center & University of Applied Sciences Leiden)

Johan den Dunnen and Ken Kraaijeveld (Leiden Genome Technology Center)

Barbara Gravendeel (Naturalis Biodiversity Center & University of Applied Sciences Leiden)

David Heijkamp (Naturalis Biodiversity Center, ICT)

Elio Schijlen (PRI WUR)

Jeroen Pijpe (University of Applied Sciences Leiden, Bioinformatics)

Rutger Vos (Naturalis Biodiversity Center, Bioinformatics)

Floyd Wittink (Generade)

Christine Minhong Zhang and Tina Xuna (Bejing Genome Institute)

# References

|  |  |
| --- | --- |
| 1. | I-Chun Pan, Der-Chih Liao, Fu-Huei Wu, Henry Daniell, Nameirakpam Dolendro Singh, Chen Chang, Ming-Che Shih, Ming-Tsair Chan, Choun-Sea Lin. April 2012. Complete Chloroplast Genome Sequence of an Orchid Model Plant Candidate: *Erycina pusilla* Apply in Tropical *Oncidium* Breeding. Date of acquirement: 2014-09-22, from<http://www.plosone.org/article/metrics/info%3Adoi%2F10.1371%2Fjournal.pone.0034738> |
| 2. | Sanger Institute. 30 Jan 2013. The Human Genome Project: a new reality. Date of acquirement: 2014-09-19, from<https://www.sanger.ac.uk/about/history/hgp/> |
| 3. | Naturalis Biodiversity Center. Publication date unknown. Anita Dirks Mulder. Date of acquirement: 2014-09-25, from<https://science.naturalis.nl/en/people/scientists/anita-dirks-mulder/> |
| 4. | Encyclopedia of Life. [Publication date](http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0034738) unknown. *Erycina pusilla* (L.) N.H. Williams & M.W. Chase. Date of acquirement: 2014-09-25, from<http://eol.org/pages/1101922/overview> |
| 5. | Pacific Biosciences. [Publication date](http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0034738) unknown. Pacific Biosciences. Date of acquirement: 2014-09-19, from<http://www.pacificbiosciences.com/> |
| 6. | Pacific Biosciences. [Publication date](http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0034738) unknown. ‘SMRT TECHNOLOGY. Date of acquirement: 2014-09-19, from<http://www.pacificbiosciences.com/products/smrt-technology/> |
| 7. | Illumina Inc. [Publication date](http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0034738) unknown. Illumina. Date of acquirement: 2014-09-19, from<http://www.illumina.com/> |
| 8. | Illumina Inc. 2007. DNA Sequencing with Solexa Technology. Date of acquirement: 2014-09-19, from<http://www.plantsciences.ucdavis.edu/bit150/2006/JD_Lecture/Lecture%201%20Databases/Solexa_DNAsequencing.pdf> |
| 9. | Orchidstra. 2012/2013. PISTILLATA gene sequence. Date of acquirement: 2014-10-01, from<http://orchidstra.abrc.sinica.edu.tw/none/citation_none.php> |
| 10. | William Staffond Noble. July 2009. A Quick Guide to Organizing Computational Biology Projects. Date of acquirement: 2014-09-28, from<http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000424> |
| 11. | Chun-lin Su, Wan-Chieh Chen, Ann-Ying Lee, Chun-Yi Chen, Yao-Chien Alex Chang, Ya-Ting Chao, Ming-Che Shih. November 2013. A Modified ABCDE Model of Flowering in Orchids Based on Gene Expression Profiling Studies of the Moth Orchid Phalaenopsis aphrodite. Date of acquirement: 2014-09-25, from<http://www.plosone.org/article/related/info%3Adoi%2F10.1371%2Fjournal.pone.0080462;jsessionid=0070E7922131D33ADCFB8053A163140A> |
| 12. | Chun-lin Su, Ya-Ting Chao, Shao-Hua Yen, Chun-Yi Chen, Wan-Chieh Chen, Yao-Chien Alex Chang, Ming-Che Shih. 2013. Orchidstra: an integrated orchid functional genomics database. Plant and Cell Physiology 54 (2):e11. [doi:10.1093/pcp/pct004](http://www.ncbi.nlm.nih.gov/pubmed/23324169) |
| 13. | Chun-lin Su, Ya-Ting Chao, Shao-Hua Yen, Chun-Yi Chen, Wan-Chieh Chen, Yao-Chien Alex Chang, Ming-Che Shih. Orchidstra: An Integrated Orchid Functional Genomics Database. Plant Cell Physiol. Feb 2013; 54(2): e11. [doi:10.1093/pcp/pct004](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3583029/) |
| 14. | Puppet Labs. Publication date unknown. What is Puppet? Date of acquirement unknown, from<http://puppetlabs.com/puppet/what-is-puppet> |
| 15. | Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler Transform. Bioinformatics, 25:1754-60. [PMID:[19451168](http://www.ncbi.nlm.nih.gov/pubmed/19451168)] |
| 16. | Joseph Fass. June 2008. Fasta1line.pl. Date of Date of acquirement unknown, from <http://wiki.bioinformatics.ucdavis.edu/index.php/Fasta1line.pl> |
| 17. | BGI. 2007,2008-2010. SOAPdenovo. Date of acquirement: 2014-10-01, from<http://soap.genomics.org.cn/about.html> |
| 18. | EMBL-EBI. 2011. Velvet. Date of acquirement:2014-10-01, from<http://www.ebi.ac.uk/~zerbino/velvet/> |
| 19. | Raymond A. 2014. ABySS. Date of acquirement: 2014-10-01, from<http://www.bcgsc.ca/platform/bioinfo/software/abyss> |
| 20. | CLC bio. 2014. CLC Genomics Workbench. Date of acquirement: 2014-10-01, from<http://www.clcbio.com/products/clc-genomics-workbench/> |
| 21. | Myers et al. (2000) A Whole-Genome Assembly of Drosophila. Science 287 2196-2204 |
| 22. | Biostars user “Sukhdepp Singh”. 2012 Question: How To Filter Mapped Reads With Samtools. Date of acquirement: 2015-01-25, from [https://www.biostars.org/p/56246/](https://www.sanger.ac.uk/about/history/hgp/) |
| 23. | Biostars user “Ian”. 2011 Question: Interpretation Of Consensus Sequence From Samtools Pileup. Date of acquirement: 2015-01-25, from [https://www.biostars.org/p/7187/](https://www.sanger.ac.uk/about/history/hgp/) |
| 24. | Encyclopedia of Genetics, Genomics, Proteomics and Informatics. Publication date: 2008. George P. Rédei. Date of acquirement: 2015-01-25, from [http://link.springer.com/referenceworkentry/10.1007%2F978-1-4020-6754-9\_9915](http://eol.org/pages/1101922/overview) |
| 25. | John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso ,David Rank, Primo Baybayan, Brad Bettman, Arkadiusz Bibillo, Keith Bjornson, Bidhan Chaudhuri, Frederick Christians, Ronald Cicero, Sonya Clark, Ravindra Dalal, Alex deWinter, John Dixon, Mathieu Foquet, Alfred Gaertner, Paul Hardenbol, Cheryl Heiner, Kevin Hester, David Holden, Gregory Kearns, Xiangxu Kong, Ronald Kuse, Yves Lacroix, Steven Lin, Paul Lundquist, Congcong Ma, Patrick Marks, Mark Maxham, Devon Murphy, Insil Park, Thang Pham, Michael Phillips, Joy Roy, Robert Sebra, Gene Shen, Jon Sorenson, Austin Tomaney, Kevin Travers, Mark Trulson, John Vieceli, Jeffrey Wegener, Dawn Wu, Alicia Yang, Denis Zaccarin, Peter Zhao, Frank Zhong, Jonas Korlach, Stephen Turner. 2009. Real-Time DNA Sequencing from Single Polymerase Molecules. Science: Vol. 323 no. 5910 pp. 133-138 [doi:10.1126/science.1162986](http://www.ncbi.nlm.nih.gov/pubmed/19023044) |
| 26. | Dave Tang. Publication date unknown. Tiki wiki: sequencing. Date of acquirement: 2014-10-01, from<http://davetang.org/wiki/tiki-index.php?page=Sequencing> |

# 

# Appendices

## List of Figures

Figure 1: Overview of Single Molecule Real Time Sequencing (SMRT).

Figure 2: Overview of Illumina dye sequencing.

Figure 3: Tool menu for launching OpenStack instance.

Figure 4: Login screen terminal work environment.

Figure 5: Overview of the created work environment.

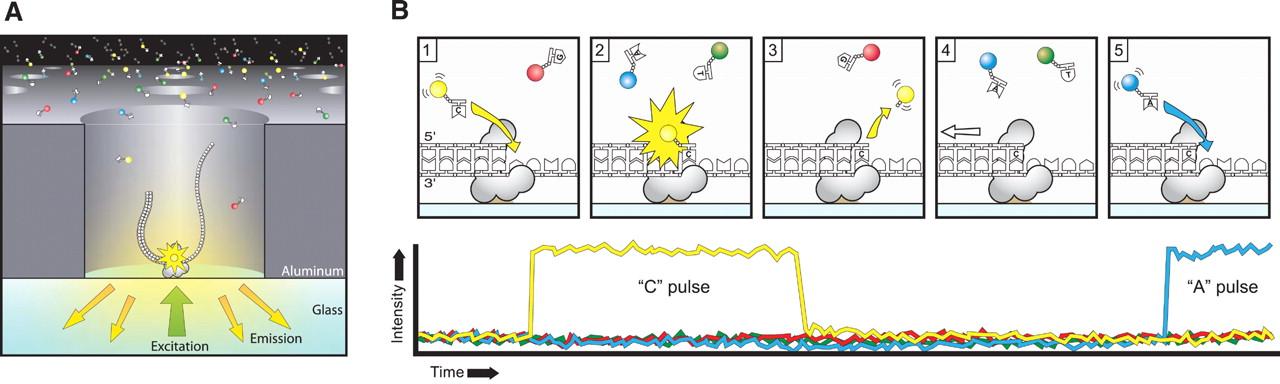


Figure 1: Overview of Single Molecule Real Time Sequencing (SMRT).

Step A: A single DNA molecule and a DNA polymerase are immobilized at the bottom of zero-mode waveguides (ZMW). Step B1: DNA polymerase starts DNA elongation. Step B2: a cytosine base with a yellow phospho-linked dye is incorporated. Excitation of the dye causes a yellow spike on the graph made by the machine. Step B3: The yellow dye is cleaved from the cytosine base. Step B4: DNA polymerase moves forward along the template. Step B5: Steps B2 through B4 repeat itself until the end of the template is reached. In this example an adenine base is incorporated which causes a blue spike on the graph.

Source: John Eid et al., 2009[27]

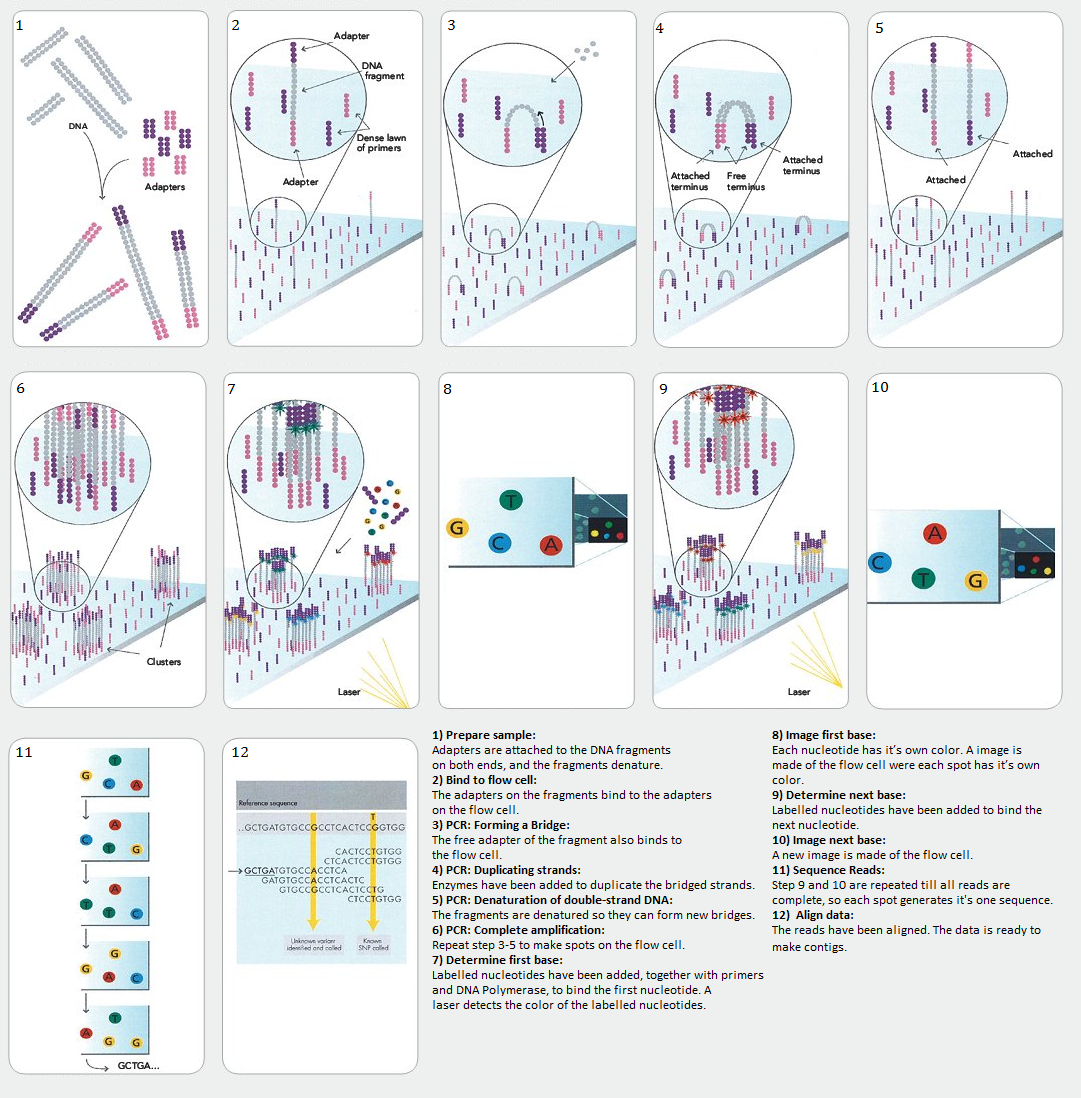


Figure 2: Overview of Illumina dye sequencing.

Source: Dave Tang[26]

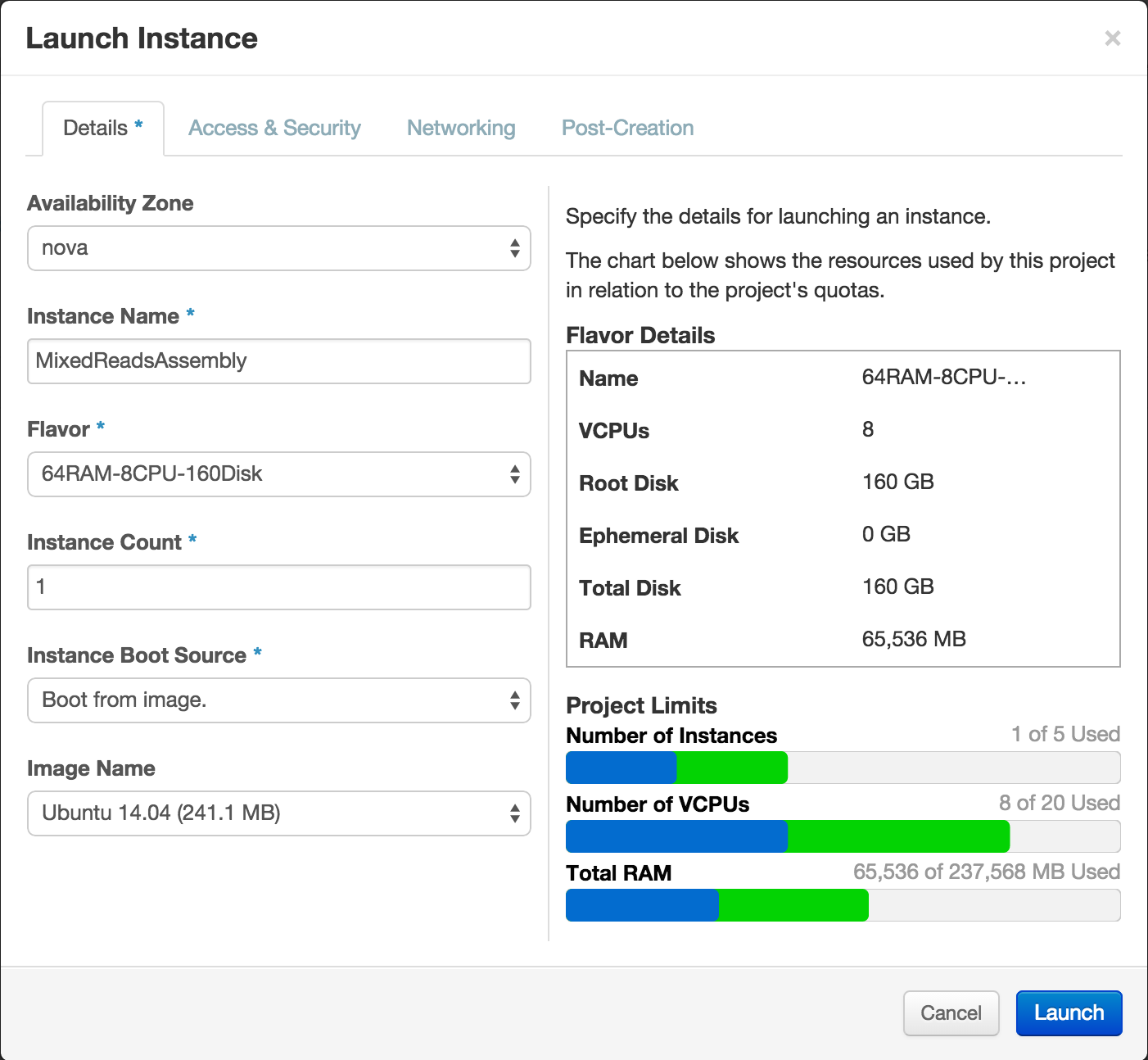


Figure 3: Tool menu for launching OpenStack instance.

This is the menu that is used to configure the OpenStack instance. Here you can specify exactly what flavour and OS you want to use, as well as specifying a post-initialization script to make sure the instance will be configured just like how you want it to be.

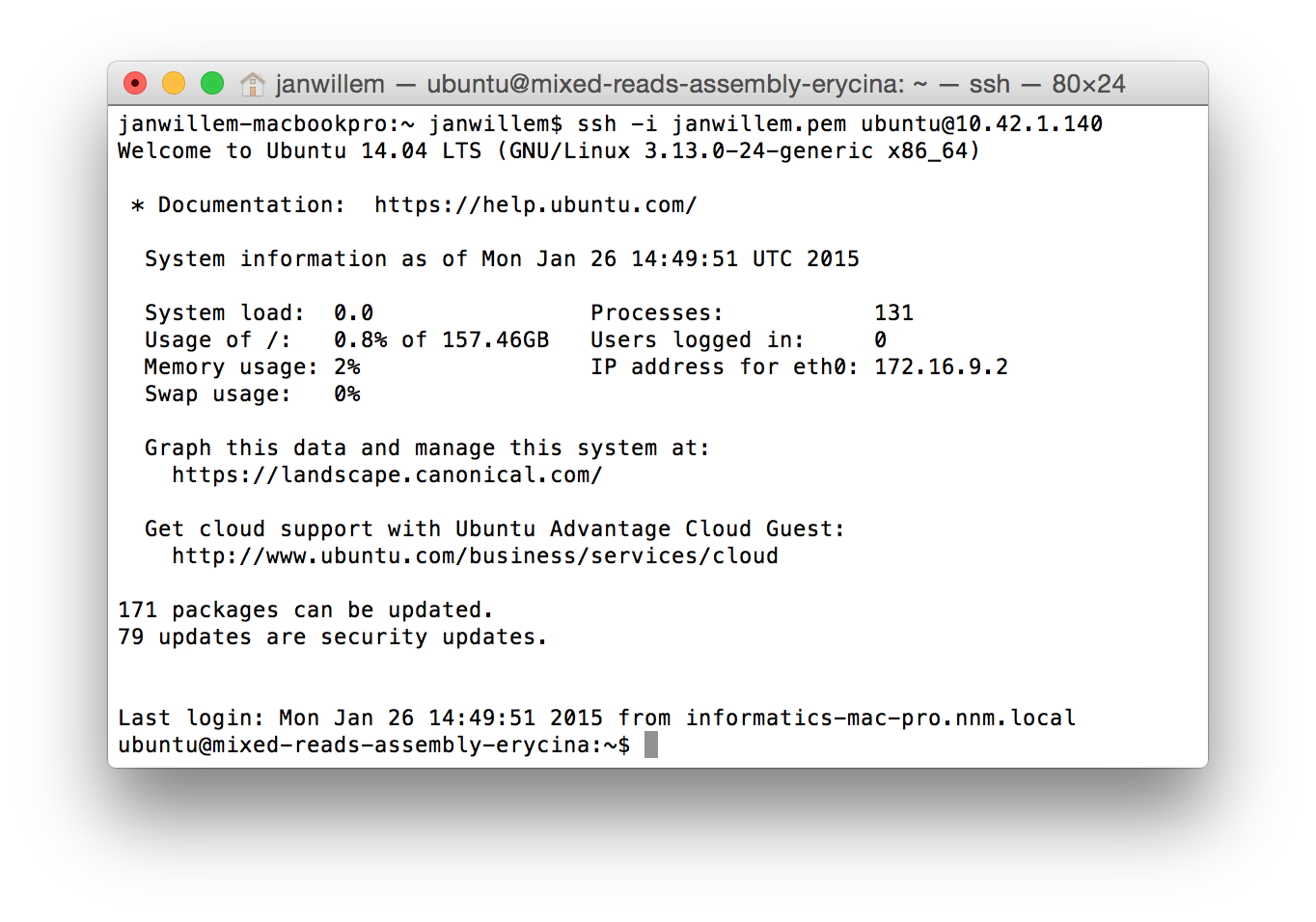


Figure 4: Login screen terminal work environment.

This is what you see when you log in into the work environment. At the top of the terminal you can see the command used to connect with the work environment. Below the command is some general information about the work environment that is always displayed on login.

## Appendix A: Puppet manifest

# update the $PATH environment variable for the Exec tasks. A

Exec {

path => [

"/usr/local/sbin",

"/usr/local/bin",

"/usr/sbin",

"/usr/bin",

"/sbin",

"/bin",

]

} end A

# Installing required packages. B

package {

"bzip2": ensure => latest;

"wget": ensure => latest;

"tar": ensure => latest;

"zlib1g-dev": ensure => latest;

"python": ensure => latest;

"python-biopython": ensure => latest;

"blast2": ensure => latest;

"ncbi-blast+": ensure => latest;

"samtools": ensure => latest;

"make": ensure => latest;

"seqtk": ensure => latest;

} end B

# set default paths for data, scripts and source code C

$username = "ubuntu"

$erycina\_dir = "/home/${username}/mixed-reads-assembly-Erycina" ##"/home/${id}/mixed-reads-assembly-Erycina"

$bin\_dir = "${erycina\_dir}/bin"

$data\_dir = "${erycina\_dir}/data"

$doc\_dir = "${erycina\_dir}/doc"

$doc\_paper\_dir = "${doc\_dir}/paper"

$results\_dir = "${erycina\_dir}/results"

$results\_assembly\_dir = "${results\_dir}/assembly"

$results\_local\_blast\_dir = "${result\_dir}/local\_blast"

$src\_dir = "${erycina\_dir}/src" end C

# create links for executables directories D

file {

$erycina\_dir:

ensure => directory,

group => $username,

owner => $username,

recurse => true;

$bin\_dir:

ensure => directory,

group => $username,

owner => $username,

recurse => true;

$doc\_dir:

ensure => directory,

group => $username,

owner => $username,

recurse => true;

$doc\_paper\_dir:

ensure => directory,

group => $username,

owner => $username,

recurse => true;

$results\_dir:

ensure => directory,

group => $username,

owner => $username,

recurse => true;

$results\_assembly\_dir:

ensure => directory,

group => $username,

owner => $username,

recurse => true;

$results\_local\_blast\_dir:

ensure => directory,

group => $username,

owner => $username,

recurse => true;

$src\_dir:

ensure => directory,

group => $username,

owner => $username,

recurse => true;

# "bwa\_link":

# path => "${bin\_dir}/bwa",

# ensure => link,

# target => "${bin\_dir}/bwa-0.7.10",

# require => Exec["unzip\_bwa"];

} end D

# command line tasks E

exec {

# install bwa

"download\_bwa":

command => "wget http://sourceforge.net/projects/bio-bwa/files/bwa-0.7.10.tar.bz2/download -O bwa.tar.bz2",

cwd => $bin\_dir,

creates => "${bin\_dir}/bwa.tar.bz2",

require => Package[ 'wget', 'tar' ];

"unzip\_bwa":

command => "tar -jxvf bwa.tar.bz2",

cwd => $bin\_dir,

creates => "${bin\_dir}/bwa-0.7.10",

require => Exec['download\_bwa'];

"make\_bwa":

command => "make",

cwd => "${bin\_dir}/bwa-0.7.10",

creates => "${bin\_dir}/bwa",

require => Exec['unzip\_bwa'];

"move\_exe":

command => "cp ${bin\_dir}/bwa-0.7.10/bwa /usr/bin/",

creates => "/usr/bin/bwa",

require => Exec['make\_bwa'];

} end E

## Appendix B: Puppet installation

#!/bin/sh A

# Cloudinit script for deploying mrbayes

set -e -x

# Git repository to clone

puppet\_source=https://github.com/naturalis/mixed-reads-assembly-Erycina.git

# Get latest puppet version end A

# Debian like B

if [ -f /usr/bin/dpkg ]

then

wget http://apt.puppetlabs.com/puppetlabs-release-stable.deb

dpkg -i puppetlabs-release-stable.deb

apt-get --yes --quiet update

apt-get --yes -o Dpkg::Options::="--force-confold" --quiet install git puppet-common ruby1.9.1 libaugeas-ruby

fi end B

# C

# Move original puppet directory

#

if [ -d "/etc/puppet.orig" ]; then

rm -rf /etc/puppet.orig

fi

mv /etc/puppet /etc/puppet.orig

#

# Fetch puppet configuration from public git repository.

#

env GIT\_SSL\_NO\_VERIFY=true git clone --recursive $puppet\_source /etc/puppet

end C

#

# Copy meta data to hiera backend directory D

#

# cloud-init json file

if [ -f /meta.js ]; then

cp /meta.js /etc/puppet/hieradata/cloud-init.json

fi

# user-data yaml file

if [ -f /user-data.yaml ]; then

cp /user-data.yaml /etc/puppet/hieradata/user-data.yaml

fi

puppet apply /etc/puppet/manifests/Testinstall\_DM2.pp end D

## Appendix C: Assembly Script

#!/bin/bash

# now using the reference from ITAG, so that A

# we might be able to crossreference regions

# with interesting results in our data (e.g. strange

# read depths, areas under selection) with

# known features

cd /media/vdb1/results/

TIME=$(date +%H:%M:%S)

mkdir $TIME

RESULTS=/media/vdb1/results/

RESULTS+=$TIME

REFERENCE=/media/vdb1/data/2014-10-31/PacBio/PRI/pacbio10longest.fasta #PacBIO

READS=/media/vdb1/data/2014-10-31/Illumina/All\_Raw\_Data/samples #illumina

CONSENSUS\_fq=/$RESULTS/consensus\_erycina.fq

CONSENSUS\_fasta=/$RESULTS/consensus\_erycina.fasta

SAMPLES=`ls $READS`

# threads for BWA align

CORES=6 End A

# recreate BWA index if not exists B

if [ ! -e $REFERENCE.bwt ]; then

echo "going to index ${REFERENCE}"

# Warning: "-a bwtsw" does not work for short genomes,

# while "-a is" and "-a div" do not work not for long

# genomes. Please choose "-a" according to the length

# of the genome.

bwa index -a bwtsw ${REFERENCE}

else

echo "$REFERENCE already indexed"

fi End B

# iterate over directories C

counter=1

for SAMPLE in $SAMPLES; do

echo "going to process sample ${SAMPLE}"

# list the FASTQ files in this dir. this should be

# two files (paired end)

FASTQS=`ls $READS/$SAMPLE/\*.fq`

# lists of produced files

SAIS=""

SAM="" End C

for FASTQ in $FASTQS; do D

# create new name

LOCALFASTA=`echo $REFERENCE | sed -e 's/.\*\///'` #Delete paths - opt\_smrtanalysis\_current\_common\_jobs\_016\_016437\_data\_filtered\_subreads.fasta

LOCALFASTQ=`echo $FASTQ | sed -e 's/.\*\///'`

OUTFILE=$READS/$SAMPLE/$LOCALFASTQ-$LOCALFASTA.sai

SAIS="$SAIS $OUTFILE"

#SAM=`echo $OUTFILE | sed -e "s/\_R.\*/-$LOCALFASTA.sam/"`

# note: we don't do basic QC here, because that might mean

# that the mate pairs in the FASTQ files go out of order,

# which will result in the bwa sampe step taking an inordinate

# amount of time

# do bwa aln if needed

if [ ! -e $OUTFILE ]; then

echo "going to align $FASTQ against $REFERENCE"

# use $CORES threads

bwa aln -t $CORES $REFERENCE $FASTQ > $OUTFILE

else

echo "alignment $OUTFILE already created"

fi

done End D

SAM="$RESULTS/$SAMPLE.sam"

# do bwa sampe (SAM\_Paired\_End) if needed E

if [ ! -e $SAM ]; then

# create paired-end SAM file

echo "going to run bwa sampe $FASTA $SAIS $FASTQS > $SAM"

bwa sampe $REFERENCE $SAIS $FASTQS > $SAM

else

echo "sam file $SAM already created"

fi End E

# do samtools filter if needed F

if [ ! -e $SAM.filtered ]; then

# -bS = input is SAM, output is BAM

# -F 4 = remove unmapped reads

# -q 50 = remove reads with mapping qual < 50

# XXX maybe increase -q?

echo "going to run samtools view -bS -F 4 -q 50 $SAM > $SAM.filtered"

samtools view -bS -F 4 -q 20 $SAM > $SAM.filtered

gzip -9 $SAM

else

echo "sam file $SAM.filtered already created"

fi End F

# do samtools sorting if needed G

if [ ! -e $SAM.sorted.bam ]; then

# sorting is needed for indexing

echo "going to run samtools sort $SAM.filtered $SAM.sorted"

samtools sort $SAM.filtered $SAM.sorted

else

echo "sam file $SAM.sorted already created"

fi

if [ $counter -eq 1 ]; then

bam1=$SAM.sorted.bam

fi

if [ $counter -eq 2 ]; then

bam2=$SAM.sorted.bam

fi

if [ $counter -eq 3 ]; then

bam3=$SAM.sorted.bam

fi

counter=$((counter+1))

done End G

#BWA MERGE H

samtools merge merged.bam bam1 bam2 bam3

# created index for BAM file if needed

if [ ! -e merged.bam.bai ]; then

# this should result in faster processing

echo "going to run samtools index $SAM.sorted.bam"

samtools index merged.bam

else

echo "BAM file index $SAM.sorted.bam.bai already created"

fi End H

# created fastq-consensus if needed I

if [ ! -e $CONSENSUS\_fq ]; then

# this should result an consensus in fasts-format.

echo "going to run samtools mpileup -uf $REFERENCE $SAM.sorted.bam | bcftools view -cg - | perl /usr/share/samtools/vcfutils.pl vcf2fq"

samtools mpileup -uf $REFERENCE merged.bam | bcftools view -cg - | perl /usr/share/samtools/vcfutils.pl vcf2fq > $CONSENSUS\_fq

else

echo "Consensus file: $CONSENSUS already created"

fi

# created fasta-consensus if needed

if [ ! -e $CONSENSUS\_fasta ]; then

# this should result an consensus in fasta-format.

echo "going to run seqtk seq -A $CONSENSUS\_fq"

seqtk seq -A $CONSENSUS\_fq > $CONSENSUS\_fasta

else

echo "Consensus file: $CONSENSUS\_fasta already created"

fi End I

## Appendix D: Fasta1line.pl

#!/usr/bin/perl  
  
# AUTHOR: Joseph Fass  
# LAST REVISED: June 2008  
#   
# The Bioinformatics Core at UC Davis Genome Center  
# http://bioinformatics.ucdavis.edu  
# Copyright (c) 2008 The Regents of University of California, Davis Campus.  
# All rights reserved.  
  
use strict;  
  
my $usage = "\nusage: fasta1line.pl <input file (fasta format)> <output file>\n\n".  
 "For each sequence, puts nt/aa all on one line following header line.\n\n";  
  
my $infile = shift or die $usage;  
my $outfile = shift or die $usage;  
  
open IN, "<$infile" or die $usage;  
open OUT, ">$outfile" or die $usage;  
  
my $header; my $sequence;  
my $firstline = 1;  
  
while (<IN>) {  
 if (m/^>/) { # recognize a header line?  
 if (!$firstline) { # output previous sequence and clear it, unless this is the first line of the file  
 print OUT $sequence."\n";  
 $sequence = "";  
 } # if  
 $firstline = 0; # we're not at the first line of the file anymore  
 print OUT; # print out header line  
 }  
 else { # not a header line? - must be sequence  
 chomp; # remove newline at end  
 $sequence .= $\_; # append additional sequence  
 } # if-else  
} # while  
print OUT $sequence."\n"; # output final sequence  
  
close IN; close OUT;

## Appendix E: PISTILLATA known gene sequence used as reference

>EPTC010697 length=887

CATTTTCTCTTAATAATCCTCTTCTCTTTTTCCCGCAAATATGGGCCGCGGCAAAATAGA

AATCAAAAGAATCGAAAACTCAACAAATCGTCAAGTTACTTTTTCCAAGAGGAGAAATGG

AATCATGAAGAAAGCGAAGGAGATCAGTGTGCTCTGTGATGCGCAAGTCTCTCTTGTTAT

ATTTTCAAGCCTTGGAAAGATGTTTGAGTATTGTAGTCCATCCACAACGTTATCGAAGAT

GTTAGAGAAGTACCAGCAGAATTCGGGGAAGAAACTGTGGGATGCGAAGCATGAGAATTT

GAGTGCGGAGATTGATCGGATCAAAAAGGAGAATGATAATATGCAGATCGAGTTGAGGCA

TTTGAAAGGGGAAGATTTGAACTCTCTTAACCCGAAAGAGCTTATTCCGATTGAAGAAGC

TCTGCAGAATGGGCTTACTGGTGTTCGGGATAAACAGATGGATTTCTTGAAGATGCTAAA

AAAGAATGAAAGAATGCTGGAAGAGGAAAATAAAAGACTTGCATACTTACTGCATCATCA

GCAATTGGCAATGGAAGGCAGCATGAGAGAACTTGATATCGGATTTCATCAGAAAGATAG

AGAATATGCAGCAGCTCAGATGCCGATGACATTTCGAGTGCAACCCATTCAACCCAACTT

GCAGGGAAACAAGTAACATTGCTAAATCTTTCCTTCAACCTGCTTTCATTCcACTTAATA

TCAATTATATAATATTAGCTTTTTAGCTGTTTCATGAGAGTATCATATACTTATGCTGAT

GATTATGATTTATATATTTGTATTGTAACTTCTTGTTGGTTGAAAATACTTTGTATCAGC

TCTGTTCAATTTGTATTTTGTTAAGGATtttctcctaattcattttc

## Appendix F: blast result of 16 day run consensus

IBLASTN 2.2.28+

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb

Miller (2000), "A greedy algorithm for aligning DNA sequences", J

Comput Biol 2000; 7(1-2):203-14.

Database: consensus\_erycina.fasta

11,682 sequences; 50,089,952 total letters

Query= EPTC010697 length=887

Length=887

\*\*\*\*\* No hits found \*\*\*\*\*

Lambda K H

1.33 0.621 1.12

Gapped

Lambda K H

1.28 0.460 0.850

Effective search space used: 42925791524

Database: consensus\_erycina.fasta

Posted date: Jan 8, 2015 10:47 AM

Number of letters in database: 50,089,952

Number of sequences in database: 11,682

Matrix: blastn matrix 1 -2

Gap Penalties: Existence: 0, Extension: 2.5

## Appendix G: blast result of 10 longest PacBio contigs consensus

BLASTN 2.2.28+

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb

Miller (2000), "A greedy algorithm for aligning DNA sequences", J

Comput Biol 2000; 7(1-2):203-14.

Database: consensus\_erycina.fasta

1 sequences; 25,523 total letters

Query= EPTC010697 length=887

Length=887

\*\*\*\*\* No hits found \*\*\*\*\*

Lambda K H

1.33 0.621 1.12

Gapped

Lambda K H

1.28 0.460 0.850

Effective search space used: 22216597

Database: consensus\_erycina.fasta

Posted date: Jan 16, 2015 9:59 AM

Number of letters in database: 25,523

Number of sequences in database: 1

Matrix: blastn matrix 1 -2

Gap Penalties: Existence: 0, Extension: 2.5

## Appendix H: blast result of one of the longest PacBio contigs consensus

BLASTN 2.2.28+

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb

Miller (2000), "A greedy algorithm for aligning DNA sequences", J

Comput Biol 2000; 7(1-2):203-14.

Database: pacbio10longest\_1.fasta

1 sequences; 40,615 total letters

Query= EPTC010697 length=887

Length=887

\*\*\*\*\* No hits found \*\*\*\*\*

Lambda K H

1.33 0.621 1.12

Gapped

Lambda K H

1.28 0.460 0.850

Effective search space used: 35320260

Database: pacbio10longest\_1.fasta

Posted date: Jan 16, 2015 12:41 PM

Number of letters in database: 40,615

Number of sequences in database: 1

Matrix: blastn matrix 1 -2

Gap Penalties: Existence: 0, Extension: 2.5

## Appendix I: blast result all PacBio contigs (Layout gefixed in word)

BLASTN 2.2.28+

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb

Miller (2000), "A greedy algorithm for aligning DNA sequences", J

Comput Biol 2000; 7(1-2):203-14.

Database: opt\_smrtanalysis\_current\_common\_jobs\_016\_016437\_data\_filtered\_subrea

ds.fasta

1,467,286 sequences; 6,639,041,197 total letters

Query= EPTC010697 length=887

Length=887

Score E

Sequences producing significant alignments: (Bits) Value

lcl|m140501\_125239\_42179\_c100643742550000001823115909101440\_s1\_p0/94839/617\_6824 ... 462 3e-127

lcl|m140501\_225401\_42179\_c100643742550000001823115909101443\_s1\_p0/100993/0\_6882 ... 390 1e-105

lcl|m140503\_225635\_42179\_c100649792550000001823116010071450\_s1\_p0/121835/0\_10550 ... 379 3e-102

lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/8830\_11833 ... 315 8e-83

lcl|m140504\_121849\_42179\_c100649792550000001823116010071454\_s1\_p0/21360/1347\_14134 ... 300 2e-78

lcl|m140501\_161148\_42179\_c100643742550000001823115909101441\_s1\_p0/122586/0\_4550 ... 261 1e-66

lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/27127\_29777 ... 191 1e-45

lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/17883\_20850 ... 156 5e-35

lcl|m140504\_153831\_42179\_c100649792550000001823116010071455\_s1\_p0/37000/0\_8004 ... 130 3e-27

lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/14883\_17833 ... 130 3e-27

lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/20897\_23913 ... 100 2e-18

lcl|m140503\_225635\_42179\_c100649792550000001823116010071450\_s1\_p0/89468/3184\_10315 ... 97.1 3e-17

lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/23957\_27086 ... 86.1 7e-14

lcl|m140504\_221935\_42179\_c100649792550000001823116010071457\_s1\_p0/130354/0\_4667 ... 62.1 1e-06

>lcl|m140501\_125239\_42179\_c100643742550000001823115909101440\_s1\_p0/94839/617\_6824

Length=6207

Score = 462 bits (250), Expect = 3e-127

Identities = 338/374 (90%), Gaps = 32/374 (9%)

Strand=Plus/Plus

Query 532 GCATCATCAGCAATTGGCAATGGAAGGCAGCATG-AGAGAACTTGAT-ATCGGATTTCAT 589

|||||||||| |||||| |||||||||||||||| ||||||| |||| ||| ||||||||

Sbjct1132 GCATCATCAG-AATTGG-AATGGAAGGCAGCATGTAGAGAAC-TGATGATC-GATTTCAT 1187

Query 590 CAGAAAGATAGAGAATATGCA-GCAGCTCAGATGCCGATGACATTTCGAGT-GCAAC-CC 646

|| ||||||||||||| |||| ||||||||||||||||||||||||||||| ||||| ||

Sbjct1188 CA-AAAGATAGAGAAT-TGCAGGCAGCTCAGATGCCGATGACATTTCGAGTGGCAACGCC 1245

Query 647 ATTCAACCCAACTTGCAGGGAAACAAGTA-ACA-TTGCTAAA-TCTTTCCTTCAACCTGC 703

||||| ||||||||||||||||||||||| ||| |||||||| || ||||||||| ||||

Sbjct1246 ATTCAGCCCAACTTGCAGGGAAACAAGTATACAGTTGCTAAAGTC-TTCCTTCAA-CTGC 1303

Query 704 TTTCATTC-C--ACTTAATA-TCAATTATATAATATTAGCTTTTT-AGCTGT-TTCATGA 757

||||||| | |||||||| ||||| || |||||||||||||| |||||| |||||||

Sbjct1304 -TTCATTCACTTACTTAATACTCAATATTA-AATATTAGCTTTTTCAGCTGTATTCATGA 1361

Query 758 GAGTATCATATACTTATGCTGATGATTATGATTTATA-TATTTGTATTGTAACTTCTTGT 816

||||| ||||||||||||||||||||||||||||||| ||||||||| |||||| |||||

Sbjct1362 GAGTA-CATATACTTATGCTGATGATTATGATTTATAATATTTGTAT-GTAACT-CTTGT 1418

Query 817 TGGTTGAAAATACTTTGTA-TCAGCTCTGTTCAATTTGTAT-TTTGTTAAGGATTTTCTC 874

||||||||||||||||||| ||||||||||||||||||||| ||||||||||||||| ||

Sbjct1419 TGGTTGAAAATACTTTGTAATCAGCTCTGTTCAATTTGTATCTTTGTTAAGGATTTT-TC 1477

Query875 CTAA-TTCATTTTC 887

||| |||||||||

Sbjct1478 GTAAATTCATTTTC 1491

>lcl|m140501\_225401\_42179\_c100643742550000001823115909101443\_s1\_p0/100993/0\_6882

Length=6882

Score = 390 bits (211), Expect = 1e-105

Identities = 331/381 (87%), Gaps = 39/381 (10%)

Strand=Plus/Plus

Query 540 AGCAATTGGCAATGGAAGGCAGCATGAGAGAACTT-G-ATAT----CGGATTTCATCAGA 593

||||||||||||||||||||||||||||||||||| | |||| |||| |||||||||

Sbjct2839 AGCAATTGGCAATGGAAGGCAGCATGAGAGAACTTGGTATATCGAGCGGA-TTCATCAGA 2897

Query 594 AAGATAGAG-AATATGCAGCAG-CTCAGATGCCGATGACATTTCGAGT-GCAAC--CCAT 648

|| ||||| |||| |||||| |||||||||||| | |||||||||| ||||| ||||

Sbjct2898 AAAGTAGAGTAATA-CCAGCAGCCTCAGATGCCGA-G-CATTTCGAGTGGCAACTGCCAT 2954

Query 649 TC--A-A--CCC-AACTTGCAGGGAAACAAGTAACATTGCT-AAATCTT-TCCTTCAACC 700

|| | | ||| |||||||||||||||||||||||||||| ||||||| ||||||||||

Sbjct2955 TCTTAGATGCCCAAACTTGCAGGGAAACAAGTAACATTGCTAAAATCTTATCCTTCAACC 3014

Query 701 TGCTTTCAT-TCCAC-TTA---ATATCAATTATATAAT--ATTAGCTTTTTAGCTGTTTC 753

|| ||||| ||||| ||| | ||||||| |||||| ||||||||||||||||||||

Sbjct3015 TG-GTTCATGTCCACTTTACTTAAATCAATTTTATAATAAATTAGCTTTTTAGCTGTTTC 3073

Query 754 ATGAGAGTATCATA-TACTTATGCTGATGATTATGATTTATA-TATTTGTA-TTGTAACT 810

|||||||||||||| ||||||||||||||||||||||||||| |||||||| ||||||||

Sbjct3074 ATGAGAGTATCATAATACTTATGCTGATGATTATGATTTATAATATTTGTAATTGTAACT 3133

Query 811 T-CTTGTTGGTTGAAAA-TACTTTGTATCAGCTCT-GTTCAATTTGTATTTTGTTAAG-G 866

| ||||||||||||||| |||||||||||| | | |||||||| ||||| |||||| |

Sbjct3134 TTCTTGTTGGTTGAAAAATACTTTGTATCACTTTTCGTTCAATTGATATTT-GTTAAGAG 3192

Query867 ATTTTCTCCTAATTCATTTTC 887

|||||||||||||||||||||

Sbjct3193 ATTTTCTCCTAATTCATTTTC 3213

>lcl|m140503\_225635\_42179\_c100649792550000001823116010071450\_s1\_p0/121835/0\_10550

Length=10550

Score = 379 bits (205), Expect = 3e-102

Identities = 343/399 (86%), Gaps = 51/399 (13%)

Strand=Plus/Minus

Query 532 GCA-TCATCAGCAATTGGCAATGGAAGGCAGC--A-T-GA--GA-GAAC--TTGATATCG 581

||| |||||||||||||||||||||||||||| | | || || |||| |||||||||

Sbjct4128 GCATTCATCAGCAATTGGCAATGGAAGGCAGCTAATTCGACCGACGAACATTTGATATCG 4069

Query 582 GATTTCATCAGAAAGATA-GAGAATATGCAGCAGC-TCAGATGCCGA-TGACATTTCGAG 638

|| ||||||||||||||| |||||||||||| ||| |||||||| || ||||||||||||

Sbjct4068 GA-TTCATCAGAAAGATATGAGAATATGCAG-AGCTTCAGATGCGGATTGACATTTCGAG 4011

Query 639 TGCAACCCATT-CAACCCAACTTGCAGGGAAA-CA-AGTAACATTGC-TAAATCTTT-CC 693

||||||||||| ||| |||||||||||||||| || ||||||||||| ||||||||| ||

Sbjct4010 TGCAACCCATTTCAAGCCAACTTGCAGGGAAAACAGAGTAACATTGCCTAAATCTTTTCC 3951

Query 694 TTCAACCT-GCTTTCATTCCACTTA---A-TATCAATT-A--TATAATATTA-GCTTTTT 744

|||||||| |||||||||||||||| | |||||||| | |||||||||| |||||||

Sbjct3950 TTCAACCTTGCTTTCATTCCACTTACTTAATATCAATTTACCTATAATATTAAGCTTTTT 3891

Query 745 AGCTGTTTCATGAGAGTATCATA--TACTTATGCTGA-TGATTATGATT-TATATATTTG 800

|| |||||||||| |||||||| |||||||||||| |||| |||||| ||||||||||

Sbjct3890 AG-TGTTTCATGACGGTATCATAAATACTTATGCTGAATGAT-ATGATTATATATATTTG 3833

Query 801 TA-T-T-GTAACTTCTTG-TTGGTTGAAAATACTT--T--GT-ATCAGCTCTGTTCAATT 851

|| | | ||||||||||| |||||| ||||||||| | || || ||||||||| ||||

Sbjct3832 TAATCTTGTAACTTCTTGATTGGTTTAAAATACTTAATCTGTTAT-AGCTCTGTT-AATT 3775

Query852 TGT-A-TT-TTGTTAA-GGATTT-TCTCCTAATTCATTT 885

||| | || ||||||| |||||| |||||||||||||||

Sbjct3774 TGTTAATTATTGTTAAAGGATTTATCTCCTAATTCATTT 3736

>lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/8830\_11833

Length=3003

Score = 315 bits (170), Expect = 8e-83

Identities = 217/237 (92%), Gaps = 13/237 (5%)

Strand=Plus/Plus

Query 1 CATTTTCTC-TTAATAATCCTCTTCTCTTTTTC-CCGCAAATATGGGCCGCGGCAAAATA 58

||||||||| |||||| |||||||||||||||| |||||||||||| ||||||| ||| |

Sbjct2016 CATTTTCTCATTAATATTCCTCTTCTCTTTTTCGCCGCAAATATGGCCCGCGGC-AAA-A 2073

Query 59 GAAATCAAAAGAATCGAAAACTCAACAAATCGTCAAGTTACTTTTTC--CAAGAGGAGAA 116

|||||| |||||||||||||||||||||||||||||||||||||||| |||||||||||

Sbjct2074 GAAATC-AAAGAATCGAAAACTCAACAAATCGTCAAGTTACTTTTTCGACAAGAGGAGAA 2132

Query 117 ATGGAATCATGAAGAAAGCGAAGGAGATCA-GTGTGCTCTGTGATGCGCAAGTCTCT-CT 174

|| |||||||||||||||||||||||||| || | ||||||||||| ||||||||| ||

Sbjct2133 -TG-AATCATGAAGAAAGCGAAGGAGATCATGTTT-CTCTGTGATGCACAAGTCTCTACT 2189

Query175 TGTTATATTTTCAAGCCTTGGAAAGATGTTTGAGTATTGTAGTCCAT-CCACAACGT 230

|||||| |||||||||||||||||||| |||||||||||||||||| |||||||||

Sbjct2190 TGTTATCCTTTCAAGCCTTGGAAAGATGATTGAGTATTGTAGTCCATTCCACAACGT 2246

>lcl|m140504\_121849\_42179\_c100649792550000001823116010071454\_s1\_p0/21360/1347\_14134

Length=12787

Score = 300 bits (162), Expect = 2e-78

Identities = 216/239 (90%), Gaps = 16/239 (7%)

Strand=Plus/Plus

Query 4 TTTCTCTTAATAAT--CCTCTTCT-CTTTTTCCCGCAAATATGGGCCGCGGCAAAATAGA 60

|||||||||| ||| |||||||| |||||||||||||||||||||||||||||||||||

Sbjct2542 TTTCTCTTAA-AATCGCCTCTTCTCCTTTTTCCCGCAAATATGGGCCGCGGCAAAATAGA 2600

Query 61 AATCAAAAGAATCGAAAA-CTCAACAAATCGTCAAGTTACTTTTTCCAAGAGGAGAAA-T 118

||| ||| |||||||||| ||||||||||||||||||||||||||||||||||||||| |

Sbjct2601 AATGAAA-GAATCGAAAAACTCAACAAATCGTCAAGTTACTTTTTCCAAGAGGAGAAAAT 2659

Query 119 GG-AATCATGAAGAAA-G-CGA-AGGAGATCAGTGTG-CTCTGTGATGCGCAAGTCTCTC 173

|| ||||||||||||| | ||| ||||||||||||| ||||||||||| | ||||||||

Sbjct2660 GGCAATCATGAAGAAAAGGCGACAGGAGATCAGTGTTTCTCTGTGATGCACCAGTCTCTC 2719

Query 174 TTGTTATATTTTCAAGCCTTGGAAAGATGTTTGAGTATT-GTAGTCCATCC-ACAACGT 230

| ||||| | ||| ||||||| ||||||||||||||||| ||||||||||| |||||||

Sbjct2720 T-GTTATCTGTTCCAGCCTTG-AAAGATGTTTGAGTATTTGTAGTCCATCCTACAACGT 2776

>lcl|m140501\_161148\_42179\_c100643742550000001823115909101441\_s1\_p0/122586/0\_4550

Length=4550

Score = 261 bits (141), Expect = 1e-66

Identities = 215/246 (87%), Gaps = 23/246 (9%)

Strand=Plus/Minus

Query 4 TTTCTC-TTAATAATCCTCTTCTCTTTTTCCCG-CAAATA-TGGGCCGCGGCAAAATA-G 59

|||||| |||||||| ||||||||||||||||| |||||| ||||||||||||||||| |

Sbjct 296 TTTCTCATTAATAAT-CTCTTCTCTTTTTCCCGACAAATACTGGGCCGCGGCAAAATACG 238

Query 60 AAATCAAAAGAATCG-AAA-ACTCAACAAATCGTCAAGTTACTTTTTCCAAGAGG-AGAA 116

|||| ||||||||| ||| ||||||||||||||||||||||||||||||||||| ||||

Sbjct 237 AAATTGAAAGAATCGTAAATACTCAACAAATCGTCAAGTTACTTTTTCCAAGAGGGAGAA 178

Query 117 A-TGGAATCATGAAGAAAGC-GAAG-GAGA-TC-AGTGTGCTCTGTG-ATGCGCAAGT-C 169

| |||| ||||||||||||| |||| |||| || | || ||||||| |||| ||||| |

Sbjct 177 AATGGA-TCATGAAGAAAGCCGAAGCGAGAATCGACGGTTCTCTGTGGATGCACAAGTTC 119

Query 170 TCT-CTTG-TTATATTTTCAA-GCCTTGGAAAGATGTTTGAGTATTGT-AGTCCATCCA- 224

||| |||| |||| ||||||| |||| ||||||||| | ||||||||| ||||||||||

Sbjct 118 TCTTCTTGCTTATCTTTTCAATGCCT-GGAAAGATGGT-GAGTATTGTTAGTCCATCCAA 61

Query 225 CAACGT 230

||||||

Sbjct 60 CAACGT 55

>lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/27127\_29777

Length=2650

Score = 191 bits (103), Expect = 1e-45

Identities = 190/228 (83%), Gaps = 22/228 (10%)

Strand=Plus/Plus

Query 6 TCTCTTAATAATCCTCTTCTCTTTTTCCC--GCAAATATGGGCCGCGGCAAAATAGAAAT 63

|||||| || || ||| ||||||| || ||| |||| ||||||||| ||||||||||

Sbjct1960 TCTCTT-AT-AT-CTC-TCTCTTTCCCCGAGGCACATAT-GGCCGCGGC-AAATAGAAAT 2013

Query 64 CAAAAGAATCGAAAACTCAACAAATCGTCAAGTTACTTTTTCCAAGAGGAGAAATGGAAT 123

| |||||||||||||||||||||||||||||| || ||| | |||||||| || | |||

Sbjct2014 C--AAGAATCGAAAACTCAACAAATCGTCAAGTGAC-TTTCCAAAGAGGAG-AAGGCAAT 2069

Query 124 CATGAAGAAAGCGAAGGAGATCAGTGTGCTCTGTGATGCGCAAGTCT-CTCTTGTTATAT 182

|| | |||||||||||||||||||||| ||||||||| ||| ||| |||| |||| ||

Sbjct2070 CA-G-AGAAAGCGAAGGAGATCAGTGTTCTCTGTGATCAGCATCTCTTCTCT-GTTA-AT 2125

Query 183 TTTCAAGCCTTGGAAAGATGTTTGAGTATTGTAGTCCATCCACAACGT 230

||||| ||| | ||||||||| ||||| ||||||||||| |||||||

Sbjct2126 CTTCAA-CCT-G-AAAGATGTT-GAGTAGTGTAGTCCATC-ACAACGT 2168

Score = 154 bits (83), Expect = 2e-34

Identities = 97/103 (94%), Gaps = 4/103 (4%)

Strand=Plus/Plus

Query 356 AGGCATTTGAAAGGGGAAGATTTGAACTCTCTTAACCCGAAAGAGCTTATTCCGATTGAA 415

|||||||||||||||||||||||||| ||||||| ||||||||||||||||||||||||

Sbjct2516 AGGCATTTGAAAGGGGAAGATTTGAA--CTCTTAAACCGAAAGAGCTTATTCCGATTGAA 2573

Query416 GAAGCTCTG-CAGAATGGGCTTACTGGTGTTCGGGATAAACAG 457

||||| ||| ||||| |||||||||||||||||||||||||||

Sbjct2574 GAAGC-CTGGCAGAAGGGGCTTACTGGTGTTCGGGATAAACAG 2615

>lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/17883\_20850

Length=2967

Score = 156 bits (84), Expect = 5e-35

Identities = 99/105 (94%), Gaps = 6/105 (6%)

Strand=Plus/Minus

Query 356 AGGCATTTGAAAGG-GGAAGATTTGA-ACTCTCTTAACCCGAAAGAGCTTATTCCGATTG 413

|||||||||||||| |||||| |||| |||||||||| ||||||||||||||||||| ||

Sbjct 411 AGGCATTTGAAAGGCGGAAGA-TTGACACTCTCTTAA-CCGAAAGAGCTTATTCCGA-TG 355

Query 414 AAGAAGCTCTGCAGAATGGGCTTACTGGTGTTCGGG-ATAAACAG 457

|||||||||||||||||||||||||||||||||||| ||||||||

Sbjct 354 AAGAAGCTCTGCAGAATGGGCTTACTGGTGTTCGGGGATAAACAG 310

Score = 89.8 bits (48), Expect = 5e-15

Identities = 60/65 (92%), Gaps = 4/65 (6%)

Strand=Plus/Minus

Query 294 AGAATTTGAGTGCGGAGATTGATCGGATCAAAAAGGAGAATGATAATATGCAG-ATCGAG 352

||||||||||||||||||||||||||||||||||||||| || | |||||||| |||| |

Sbjct 576 AGAATTTGAGTGCGGAGATTGATCGGATCAAAAAGGAGA-TG-TCATATGCAGTATCG-G 520

Query 353 TTGAG 357

|||||

Sbjct 519 TTGAG 515

>lcl|m140504\_153831\_42179\_c100649792550000001823116010071455\_s1\_p0/37000/0\_8004

Length=8004

Score = 130 bits (70), Expect = 3e-27

Identities = 97/108 (90%), Gaps = 9/108 (8%)

Strand=Plus/Plus

Query 356 AGGCATTTGAAAGGGGAAGATTTG-AACTCTCTTAACCCGAAAGAGCTTATTCCGATTGA 414

|||||||| ||| ||||||||||| ||| ||| |||||||||||||||||||||||||||

Sbjct5873 AGGCATTT-AAA-GGGAAGATTTGAAACGCTC-TAACCCGAAAGAGCTTATTCCGATTGA 5929

Query 415 AGAAGCTC-TGC-AGAATG-GGCTTA-CTGGTGTTC-GGGATAAACAG 457

|||||||| ||| ||||| |||||| ||||||||| |||||||||||

Sbjct5930 AGAAGCTCCTGCCAGAATATGGCTTAACTGGTGTTCCGGGATAAACAG 5977

>lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/14883\_17833

Length=2950

Score = 130 bits (70), Expect = 3e-27

Identities = 93/102 (91%), Gaps = 9/102 (9%)

Strand=Plus/Plus

Query 358 GCATTTGAAAGGGGAAGATTTGAACTCTCTTAACCCG--AAAGAGCTTATTCCGATTGAA 415

|||||||||| |||||||||||||||||| ||||||| |||||||||| ||||||||||

Sbjct2553 GCATTTGAAA-GGGAAGATTTGAACTCTC-TAACCCGAAAAAGAGCTTA-TCCGATTGAA 2609

Query 416 GAAGCTCTGCAGAATGGG-CTTAC-TGG-TGTTCGGGATAAA 454

|||||||||||||||||| ||||| ||| ||||||| |||||

Sbjct2610 GAAGCTCTGCAGAATGGGGCTTACCTGGGTGTTCGG-ATAAA 2650

Score = 113 bits (61), Expect = 3e-22

Identities = 64/65 (98%), Gaps = 1/65 (2%)

Strand=Plus/Plus

Query 294 AGAATTTGAGTGCGGAGATTGATCGGATCAAAAAGGAGAATGATAATATGCAGATCGAGT 353

||||||||||||||||||||||||| ||||||||||||||||||||||||||||||||||

Sbjct2413 AGAATTTGAGTGCGGAGATTGATCG-ATCAAAAAGGAGAATGATAATATGCAGATCGAGT 2471

Query 354 TGAGG 358

|||||

Sbjct2472 TGAGG 2476

>lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/20897\_23913

Length=3016

Score = 100 bits (54), Expect = 2e-18

Identities = 64/68 (94%), Gaps = 4/68 (6%)

Strand=Plus/Plus

Query 229 GTTATCGAAGATGTTAGAGAAGTACCAGCAGAATTCGGGGAAGAAACTGTGGGATGCGA- 287

|||||||||||||||||||||||||||||||||||||| |||||||||||| ||||||

Sbjct2341 GTTATCGAAGATGTTAGAGAAGTACCAGCAGAATTCGG--AAGAAACTGTGG-ATGCGAC 2397

Query 288 AGCATGAG 295

||||||||

Sbjct2398 AGCATGAG 2405

Score = 84.2 bits (45), Expect = 3e-13

Identities = 45/45 (100%), Gaps = 0/45 (0%)

Strand=Plus/Plus

Query 488 GAAAGAATGCTGGAAGAGGAAAATAAAAGACTTGCATACTTACTG 532

|||||||||||||||||||||||||||||||||||||||||||||

Sbjct 2936 GAAAGAATGCTGGAAGAGGAAAATAAAAGACTTGCATACTTACTG 2980

>lcl|m140503\_225635\_42179\_c100649792550000001823116010071450\_s1\_p0/89468/3184\_10315

Length=7131

Score = 97.1 bits (52), Expect = 3e-17

Identities = 88/103 (85%), Gaps = 12/103 (12%)

Strand=Plus/Minus

Query 356 AGGCATTTGAAAGGGGAAGATTTGAACTCTCTTAACCCGAAAGAGCTT-ATTCCGATT-G 413

|||||||||||||||||||||||||||| | |||||| ||| |||||| |||| |||| |

Sbjct 671 AGGCATTTGAAAGGGGAAGATTTGAACTAT-TTAACC-GAA-GAGCTTTATTC-GATTTG 616

Query 414 AAGAAGCTCTGCAGAATGGGCTTACTGGTGTT-CGGGATAAAC 455

|||| ||| ||| || |||| ||| || |||| ||||||||||

Sbjct 615 AAGA-GCT-TGC-GA-TGGGTTTA-TGTTGTTTCGGGATAAAC 578

>lcl|m140502\_121921\_42179\_c100643742550000001823115909101447\_s1\_p0/20160/23957\_27086

Length=3129

Score = 86.1 bits (46), Expect = 7e-14

Identities = 64/71 (90%), Gaps = 7/71 (10%)

Strand=Plus/Minus

Query 229 GTTATCGAAGATGTTAGAGAAGTACCAGCAGAATTCGGGGAAGAAACTGTG-GGAT-GC- 285

|||||||||||||||||||||||||||||||| |||||||| ||| ||||| |||| ||

Sbjct 702 GTTATCGAAGATGTTAGAGAAGTACCAGCAGA-TTCGGGGA-GAA-CTGTGCGGATTGCT 646

Query 286 G-AAGCATGAG 295

| |||||||||

Sbjct 645 GGAAGCATGAG 635

>lcl|m140504\_221935\_42179\_c100649792550000001823116010071457\_s1\_p0/130354/0\_4667

Length=4667

Score = 62.1 bits (33), Expect = 1e-06

Identities = 33/33 (100%), Gaps = 0/33 (0%)

Strand=Plus/Minus

Query 456 AGATGGATTTCTTGAAGATGCTAAAAAAGAATG 488

|||||||||||||||||||||||||||||||||

Sbjct 2709 AGATGGATTTCTTGAAGATGCTAAAAAAGAATG 2677

Lambda K H

1.33 0.621 1.12

Gapped

Lambda K H

1.28 0.460 0.850

Effective search space used: 5644083363336

Database: opt\_smrtanalysis\_current\_common\_jobs\_016\_016437\_data\_filtered\_subrea

ds.fasta

Posted date: Jan 16, 2015 1:00 PM

Number of letters in database: 6,639,041,197

Number of sequences in database: 1,467,286

Matrix: blastn matrix 1 -2

Gap Penalties: Existence: 0, Extension: 2.5

## Appendix J: Overview of different datasets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Sequencer** | **File format** | **Size** | **Description** |
| opt-smrtanalysis-current-common-jobs-016-016437-data-filtered\_subread.fasta | Pacbio | Fasta | 6.726.965 kB (= 6.41 GB) | Pacbio scaffolds |
| 111210\_I595\_FCC07FEACXX\_L4\_SZAXPI002374-11\_1.fq.qz | Illumina | FastQ (Qzipped) | 13.105.994 kB (= 12.4 GB) | Illumina contigs |
| 111210\_I595\_FCC07FEACXX\_L4\_SZAXPI002374-11\_2.fq.qz | Illumina | FastQ (Qzipped) | 14.278.130 kB (= 13.6 GB) | Illumina contigs |
| 111217\_I123\_FCD0HJEACXX\_L6\_SZAXPI002377-12\_1.fq.qz | Illumina | FastQ (Qzipped) | 18.915.129 kB (=18.0 GB) | Illumina contigs |
| 111217\_I123\_FCD0HJEACXX\_L6\_SZAXPI002377-12\_2.fq.qz | Illumina | FastQ (Qzipped) | 19.226.212 kB (=18.4 GB) | Illumina contigs |
| s\_2\_1\_sequence (paired) mapping.clc | Illumina | Clc | 119,396 kB (= 116.6 MB) | *Erycina pusilla* chloroplast |
| s\_2\_1\_sequence (paired) un-mapped reads(paired).clc | Illumina | Clc | 3.712.457 kB (= 3.5 GB) | *Erycina pusilla* chloroplast |
| s\_2\_1\_sequence (paired) un-mapped reads.clc | Illumina | Clc | 7.172 kB (= 7 MB) | *Erycina pusilla* chloroplast |
| s\_2\_1\_sequence (paired) summary report.clc | Illumina | Clc | 12 kB | *Erycina pusilla* chloroplast |
| Orchid\_soapk23.contig | Illumina | Contig | 1.308.798 kB (= 1.24 GB) | HiSeq analysis  Contig file |
| orchid\_soapk23.contig\_blastdb.00.nhr | Illumina | contig\_blastdb.00.nhr | 1.048.576 kB (= 0.99 GB) | HiSeq analysis  Blast database header file |
| orchid\_soapk23.contig\_blastdb.00.nin | Illumina | contig\_blastdb.00.nin | 129.804 kB (= 126 MB) | HiSeq analysis  Blast database index file |
| orchid\_soapk23.contig\_blastdb.00.nsq | Illumina | contig\_blastdb.00.nsq | 79.785 kB (= 76.9 MB) | HiSeq analysis  Blast database sequence file |
| orchid\_soapk23.contig\_blastdb.01.nhr | Illumina | contig\_blastdb.01.nhr | 527.231 kB (= 514 MB) | HiSeq analysis  Blast database header file |
| orchid\_soapk23.contig\_blastdb.01.nin | Illumina | contig\_blastdb.01.nin | 65.023 kB (=63.4 MB) | HiSeq analysis  Blast database index file |
| orchid\_soapk23.contig\_blastdb.01.nsq | Illumina | contig\_blastdb.01.nsq | 118.068 kB (=115 MB) | HiSeq analysis  Blast database sequence file |
| Orchid\_soap23.scafSeq | Illumina | scafSeq | 241.679 kB (=236 MB) | HiSeq analysis  Scaffold sequence file |
| Orchid\_soap31.contig | Illumina | Contig | 1.109.942 kB (=1.05GB) | HiSeq analysis  Contig file |
| Orchid\_soap31.scafSeq | Illumina | scafSeq | 310.647 kB (= 303 MB) | HiSeq analysis  Scaffold sequence file |
| s\_2\_1\_sequence.txt | Illumina | Txt | 6.000.264 kB (= 5.72 GB) | HiSeq analysis |
| s\_2\_2\_sequence.txt | Illumina | Txt | 6.044.123 kB (= 5.76 GB) | HiSeq analysis |
| **Name** | **Sequencer** | **File format** | **Size** | **Description** |
| Orchidstra[11,12] | - | Fasta | - | Database containing 83150 sequences of *Erycina pusilla*. Information for each sequence (may) include Fasta file, BLAST results, GO terms, KEGG information, Pfam information, expression profiles, homology and miRNA information. |