UPPSALA
UNIVERSITET

UPTEC X 15 034
Examensarbete 30 hp
Maj 2016

Increasing bioinformatics in third world countries

Studies of S.digitata and P.polymyxa to further bioinformatics in east Africa

Isak Sylvin

UPPSALA UNIVERSITET

Degree Project in Molecular Biotechnology

Masters Programme in Molecular Biotechnology Engineering,
Uppsala University School of Engineering

UPTEC X 15 034	**Date of issue 2016-05**

Author

Isak Sylvin

Title (English)

Increasing bioinformatics in third world countries - Studies of *S. digitata* and *P. polymyxa* to further bioinformatics in east Africa

Abstract

Despite an increase of biotechnical studies in third world countries, the bioinformatical side is largely lacking. In this paper we attempt to further the bioinformatical capabilities of east Africa. The project consisted of two teaching segments for east African doctorates, one as part of an academic workshop at ILRI, Kenya, and one in a small class at SLU, Sweden. The project also included the generation of two simple to use bioinformatical pipelines with the explicit aim to be reused by novice bioinformaticians from the very same region. The viability of the piplines were verified by generating transcriptional expression level differences for Paenibacillus polymyxa strain A26 and whole genome annotations for Setaria digitata. Both pipelines may have some merit for the collaborative effort between ILRI and SLU to annotate Eleusine coracana, a draught resilient crop, the annotation of which may save lives. The teaching material, source code for the pipelines and overall teaching impression have been included in this paper.

Keywords

Bioinformatics, pipeline, Setaria digitata, Paenibacillus polymyxa, Eleusine coracana, annotation, expression level, east Africa, eBioKit, MAKER, Cufflinks, third world country, genome, transcriptome

Supervisors

Erik Bongcam-Rudloff
SLU Swedish University of Agriculture

Scientific reviewer

Ola Spjuth
Uppsala University

Project name	Sponsors
Language **English**	Security
ISSN 1401-2138	Classification
Supplementary bibliographical information	Pages **51**

Biology Education Centre Biomedical Center Husargatan 3, Uppsala
Box 592, S-751 24 Uppsala Tel +46 (0)18 4710000 Fax +46 (0)18 471 4687

Populärvetenskaplig sammanfattning

Genmodifierade grödor, GMO, är vid skrivande stund fortfarande ett hett diskussionsämne. För inte särskilt många år sedan fullkomligt blomstrade debatter kring de etiska frågorna i ämnet. Frågor som ”Kan människor bli sjuka av genmodifierad mat?” och ”Kommer GMO att konkurrera ut den lokala faunan?” är bara några få av de många frågor som allmänheten hade gällande detta nya sätt att påverka grödor. Utifrån tonläget i många av dom debatter som uppkom, framstod den allmänna opinionen som starkt negativt inställd gentemot GMO.

Efter några år av dvala återskapades oron över GMO genom att en ny fråga fick liv, nämligen ”*Vad är konsekvenserna av att stora företag äger alla rättigheter till grödorna vi äter?”.* Allmänheten var lika negativ som innan och man behöver i skrivandes stund inte sträcka sig särskilt långt för att hitta exempel på detta. En av de större diskussionerna rörde den (fiktiva) uppsjö av stämningsansökningar som Monsanto, ett ledande företag inom GMO och bekämpningsmedel, utfärdat mot de bönder som missbrukat villkoren vid användning av deras frön genom att låta dem föröka sig. Enligt flera konspirationsteorier hade Monsanto också, i egenskap av att vara ett stort företag, betalat den vetenskapliga sfären för att skapa missvisande studier som påvisade hur biologiskt harmlösa GMO var.

Samtidigt på andra sidan världen så finns det fortfarande många länder, främst i Afrika och Asien, vars befolkning ofta har problem både med näringsbrist och svält [(1)]. Båda dessa problem beror på, i min mening, ensidig agrikultur som inte täcker näringsbehovet och som väldigt lätt slås ut av torka. Att berika de odlade grödorna med gener som tillför antingen spårämnen eller ger ökad resistans mot torka är i dagsläget en av de mest lovande lösningarna för att minska dessa problem.

Den negativa opinionen kring GMO är dock ett stort hinder. Misstron för stora företag och GMO är så pass hög att ledare av utsatta nationer har valt att inte låta deras invånare odla och konsumera GMO [(2)], ens i situationer där alternativet mycket väl kunnat leda till kraftig hungersnöd. För att GMO ska få någon form av fäste krävs det således att den lokala befolkningen kan ta fram grödorna under sina premisser, vilket skulle erbjuda en helt ny nivå av transparens för ledarna och befolkningen av länderna.

I dagsläget är många u-länder kapabla att göra sina egna biotekniska studier. Trots det så krävs det fortfarande mer resurser. Nästa steg är att se till att u-länder kan göra sin egen datordrivna analys. I skrivandes stund är det normala att provtagningen för ett projekt görs av den lokalbefolkningen i landet. Analysen görs sedan av västerländska företag eller institutioner. Detta medför att forskningen blir riktad utefter de västerländska deltagarnas villkor och värderingar snarare än den lokalbefolkningens.

Vi tror att genom att ge den lokala befolkningen de verktyg de behöver för att kunna göra den datorstyrda analysen på egen hand så kommer det bidra till ökad välfärd i många u-länder. Ländernas opinion kommer att svänga till att vara mer välkomnande till GMO, minska de lokala närings- och svältproblemen, och göra länderna mer oberoende jämtemot stora agrikulturföretag.

Detta projekt är ett av många i Erik Bongcam-Rudloffs grupp som alla bidragit till att öka de bioinformatiska resurserna i u-länder. Det mest noterbara involverade att distribuera så kallade *eBioKits* [(3)], en serverlösning som gjorde det möjligt för mottagarna att använda många typiska bioinformatiska verktyg utan regelbunden tillgång till internet. Just detta projekt ämnade att utbilda forskare i u-länder, främst i östra Afrika, i bioinformatik för att ge dem större möjligheter till att själva göra den bioinformatiska analys som krävdes. Lösningarna som togs fram var menade att kräva minimala mängder internettillgänglighet, programmeringskunskaper och IT-kunskaper.

Table of Contents

Glossary and abbreviations

Ab initio	Gene predictors that use pattern recognition and training rather than comparing with known targets
BLAST	BLAST (Basic Local Alignment Search Tool) is a well-established tool for comparison of primary biological sequence information.
e-val	The amount of false positive hits a database search (typically BLAST) will yield simply based on the exclusion criteria in relation to the size of the database.
eBioKit	A local computational cluster for bioinformatical purposes, delivered to developing regions such as Kenya, to support their bioinformatical needs.
GEO	Gene Expression Omnibus; a database repository of high throughput gene expression data.
GMO	Genetically Modified Organism
HPC	High-performance computer
ILRI	International Livestock Research Institute. A university located in Nairobi, Kenya.
N50	The sum of contigs of this length or longer make up at least half the length of sequence data for the entire set. Inversely, the sum of all contigs this length or shorter is also equal to at least half half the length of sequence data for the entire set.
NCBI	The National Center for Biotechnology Information
RAST	RAST (Rapid Annotation using Sybsystem Technology) is a fast annotation pipeline that requires minimal setup time.
Repeat Masking	Flagging repeat rich sections of the genome to be ignored by gene predictors.
RNA-Seq	RNA Sequencing, also called transcriptome shotgun sequencing is a technology that uses next-generation sequencing to reveal a snapshot of RNA.
SLU	Swedish University of Agriculture. A university located in Uppsala, Sweden.
SOP	Standard Operating Procedure
SRA	Sequence Read Archive
UPPMAX	UPPMAX (Uppsala Multidisciplinary Center for Advanced Computational Science) is a resource of high-performance computers and large-scale storage located at Uppsala University.

Introduction

This paper encompasses one of several projects to improve the welfare of developing countries by increasing their bioinformatical capabilities. This project in question was a joint collaboration between the Swedish University of Agriculture, SLU, and the International Livestock Research Institute, ILRI. For the project east African academics were invited to ILRI, Kenya, to attend bioinformatical lectures and workshops held by representatives from all over the world. In addition to this select African academics were invited to SLU, to participate in more advanced bioinformatical training.

The bioinformatical solutions that were taught during these instances were also concurrently laying the bioinformatical groundwork necessary for the collaborative project between ILRI and SLU to annotate the genome of the African finger millet, *Elusine coracana.* At the time of the project African academics had already both cultivated and sequenced a large portion of the plant and were awaiting the bioinformatical analysis necessary for publication, as well as further research. It was necessary for the bioinformatical pipelines to be designed to be general, efficient and easy to understand to a degree that was much higher than it was for typical bioinformatics. In order to achieve this the pipelines were not only developed with that mindset, but also applied to other organisms as a means of verification.

Two pipelines were constructed. One pipeline served to annotate the genome of the parasitic roundworm *Seteria digitata.* The other was constructed to analyze the transcriptomic differences of the A26 strain of *Paenibacillus polymyxa* with and without stress. By only relying on free, contemporary and open-source applications the project aimed to develop pipelines that required both minimal costs as well as bioinformatical expertise. The pipelines were also constructed with reusability in mind, with the hope that they could be reused with minimal modifications for other bioinformatical projects by aspiring academics in developing countries.

Background

East Africa is a region that consists of 20 countries [(4)]. The majority of these countries have a long history of instability. Conflict, famine and aggressive western colonization [(1)] are just a few of the issues that these nations have faced semi-regularly and are thus heavily influenced by. The issues are not just of historical significance, but are rather still very real as to this day. For instance, recent incidents such as the Somali civil war and the internal political-ethnic conflict in South Sudan are both ongoing and add to the turmoil of the region. Despite this the region houses academic resources, which despite of the instability are notably eager to use science in various ways to find solutions to the local problems. Perhaps it is because the technologies we take for granted only recently became more available to them, perhaps it is because the issues are affecting the very area they live in.

One such problem is the looming threat of hunger, malnutrition and even starvation from poor harvests [(1)]. The current agriculture of most countries in the region can support the population, but it is a fragile system. The supply very narrowly satisfies the demand under ideal circumstances. As such when circumstances change, like in the form of extended droughts, many go hungry. By talking to local Kenyans it became clear that food scarcity happened so often and unexpectedly that it was considered a natural part of life to be considerate of it. In spite of the political instability and famines, many of the countries in the region host a limited but nonetheless dedicated group of academics.

It is nowadays typical for western scientists to ask the local academics to cultivate samples for biotechnical studies rather than moving the entire research team onsite. In some cases the local academics are

also assigned to perform some, if not all, of the biotechnical work required. This is one of many factors that have led to the establishment of more biotechnical research groups than one would typically expect given the economic turbulence of the area. On the other hand the bioinformatical side of these regions is largely lacking. One of the reasons for this could be the necessity of good infrastructure for modern bioinformatics. The electrical grid is unstable and prone to sudden outages. The internet, a resource we take for granted, is a luxury provided by slow satellite connections; thus making any web based applications near useless. Finally local high speed computational clusters are not only a costly investment by themselves but also require the expert bioinformaticians that maintain them to reside in the area.

In an effort to alleviate both the hunger issues as well as the lacking bioinformatical knowledge in the region a joint collaboration between ILRI and SLU was formed. This exchange was established to, amongst other things, develop more draught resistant agricultural plants as well as strengthen the bioinformatical side of the region. A key component of both these goals was to train east African academics in both the use and implementation of bioinformatical pipelines on local computational clusters.

The mobile computational cluster

Erik Bongcam-Rudloff, whom was the supervisor of this project, and his group have had previous collaborative efforts with various universities situated in developing countries. One of these projects was the implementation the server-side solution known as eBioKit [3]. In brief, eBioKit [3] is a local computational cluster delivered to developing regions, such as Kenya, to support their bioinformatical needs. It is a self-contained, portable, UNIX server which also comes pre-installed with up-to-date bioinformatical software and databases. An example of such application would be BLAST [23] and the related databases necessary to properly run it.

Prior to eBioKit [3] implementation many academics in developing countries would be restricted to bioinformatical projects that could be analyzed within a viable timeframe on local personal computers. Personal computers with approximately ten year old hardware and internet connections that could only be described as underwhelming by western standards. Access to the proper tools did however highlight another problem, namely that there was a serious lack of bioinformatical expertise in these regions. The computer clusters needed to be administered, but more importantly, very few academics knew how to take advantage of them. The alternative of relying on western academics for the bioinformatical support would only marginally differentiate from simply delocalizing the bioinformatical analysis. It was thus necessary to properly formulate relatively simple bioinformatical pipelines and then teach them to the local academics so they could perform their own independent bioinformatical analysis.

The impact of annotating *E. coracana*

As previously mentioned the populations of many eastern African nations, including Kenya, suffer from an ongoing threat of famine [5]. The effects of this is not only felt when a famine actually occurs, but is also a great source of uncertainty and stress even when the harvests are good. One possibly way to alleviate the impact and frequency of poor harvests would be to introduce properties from the African finger millet, *E. coracana,* to the otherwise corn-based agriculture. *E. coracana* is a traditionally east African cereal which is rich in methionine, calcium and iron. *E. coracana* is however most notable for its draught resistance that likely stems from its African heritage. It is however impossible to integrate *E. coracana* into the agriculture in its current form as every individual plant provides very small yields. A possible solution would be to either alter the corn, *Zea mays,* with the draught resistant capacities of *E. coracana*, or modify *E.coracana* to increase its yields.

The idea to introduce draught resistant crops to African agriculture in some form is not a novel one. However in the past a lack of funding has postponed its realization indefinitely. There is very little economical gain in researching this solution for western industries as most countries that suffer from draught related problems are almost exclusively third world countries. There is in other words almost no market in industrialized nations, and third world countries are per definitions poor. One could therefore reasonably assume that if the research was not performed in a developing nation, it would not be performed at all.

Curing elephantiasis by researching *S. digitata*

In order to verify that the annotation pipeline performs well enough despite being relatively simple, it was suggested to have it verified by annotating another organism. Thus the pipeline will be tested by annotating and evaluating the results for the much smaller *S. digitata* genome. Annotating *S. digitata* does however come with its own biological merits.

Firstly this primarily bovine filarial parasite can cause fatal paralysis to the host organism. The parasite has also been reported to infect goats, sheep and horses; meaning that not only cattle farmers are at risk. This in turn may be of dire consequences to the farmers as *S. digitata* is indigenous to Sri Lanka, a region where the farmers' profit margins are mostly slim. Secondly *S. digitata* shares several similarities with the nematode *Wuchereria bancrofti.* They both share the same phylum, Nematoda, and use a similar intermediate vector. Both rely on mosquitos. *S. digitata* uses the mosquito *Aedes aegypti* whilst *W. bancrofti* uses the mosquito *Anopheles culifaciens.* Based on these similarities we decided that *S. digitata* would perform well as a model organism to better understand *W. bancrofti.*

W. bancrofti is notable for causing human lymphatic filariasis, also known as elephantiasis [(8)]. This has been identified as the second leading cause of long-term and permanent disability [(9)]. Although it causes little direct mortality, it results in the development of profound debilitating morbidity. As *W. bancrofti* almost exclusively infects residents of tropical third world countries, whom seldom have alternatives to physical labor, the diseased is very unlikely to be able to support himself ever again. This in turn has an immense socio- economic impact on the affected individuals and their respective families.

An estimated 128 million people worldwide are currently infected or diseased with lymphatic filarial organisms. Of these *W. bancrofti* is expected to be responsible for approximately 115 million cases [(10)] or 89 percent. This value can be contrasted to those who are infected by second most common carrier *Brugia malayi. B. malayi* is suspected to have infected 13 million individuals, 10 percent of all known cases. Although *W. bancrofti* is indisputably the most frequent carrier of the disease, very little is known about the parasite's molecular biology, biochemistry and immune mechanisms.

As of writing there exists no vaccine against human lymphatic filariasis. There is however two drugs currently available for treating the disease; Diethycarbazepine and Ivermectin [(8)]. Given the number of infected and socio-economic impact the disease causes, one can conclude there is a large discrepancy between the supply and demand. There are two major reasons as to why this discrepancy exists.

First and foremost the research costs for developing treatments against human lymphatic filariasis caused by *W. bancrofti* is very high. The parasitic material suffers from paucity since *W. bancrofti* cannot be maintained in a laboratory environment. Researchers must thus either make frequent trips between central Africa and their own lab to continually harvest fresh specimens, or alternatively set-up a lab in central Africa and perform much of the research on-site. Secondly there is very little financial incentive

for international pharmaceutical companies to invest in research to identify new drug development targets relating to *W. bancrofti*. Most of the individuals infected with lymphatic filarial organisms are residents of third world countries, whose own personal income mimic that.

The research costs of analyzing *S. digitata* pales in comparison to analyzing *W. bancrofti* and may prove a viable alternative. The organism is native to Asian third world countries, but is easily harvested en-masse from the cattle.

What is a pipeline and why do we use them?

One of the more common issues when starting with bioinformatics is to not know where to start. The magnitude of available bioinformatical applications is astounding, and may deter potential scientists. For the most part each of these applications are designed to solve one specific key step in the bioinformatical analysis. In addition to this each application is typically developed independently from any others, thus quickly becoming the leading cause of compatibility issues. Generally a bioinformatician is often finding themselves in a position where they have to learn a specific solution, translate the output to input for another solution, note any shortcomings of the algorithms used before moving on to another solution. As an average bioinformatical workflow require up to 10 different software solutions errors are both time consuming and bound to happen. This is further elevated by the constantly increasing amount of bioinformatical data for projects that were considered impossible in the past.

One way to alleviate these problems is to rely on predefined bioinformatical pipelines. Both MAKER [12] and the Cufflinks suite [11] are considered to be this to varying degrees. These two software solutions have been designed to incorporate several applications into a bundle to minimize the required intermediary scripting necessary to perform bioinformatical analysis. Both software solutions are also designed to clearly suggest which step follows which, as to make it easier for aspiring bioinformaticians to not get dumbstruck.

More descriptively MAKER [12] is an easy-to-configure genome annotation pipeline with minimal inputs. MAKER [12] allows participants of small genome projects to effectively annotate their genomes and to create genome databases. MAKER [12] identifies repeats, aligns ESTs and proteins to a genome, produces ab initio gene predictions and produces them into gene annotations. MAKER [12] can also be trained on outputs of preliminary runs to automatically retrain its gene prediction algorithm. Its outputs can be directly loaded into the visualizer Web Apollo [13], produced by the same developer. [14]

The Cufflinks suite [11] assembles transcripts, estimates their abundances, and tests for differential expression and regulation in RNA-seq samples. It accepts aligned RNA-seq reads and assembles the alignments into a parsimonious set of transcripts. Cufflinks then estimates the relative abundances of these transcripts based on how many reads support each one. [15]

Methodology

MAKER overview

Maker [12] uses a seven step workflow to produce its results (Figure 1). In the first half of the workflow Maker [12] first masks regions of repeating segments of the genome from being analyzed. It then runs ab initio gene prediction software. Following this it uses algorithms that rely on supplied EST and protein evidence from data from related organisms to the target to make an additional gene prediction.

In the second half of the workflow MAKER [12] fine tunes the resulting gene predictions before using them to train the more complex predictors. Finally quality metrics are generated from the session and low quality gene predictions are further filtered.

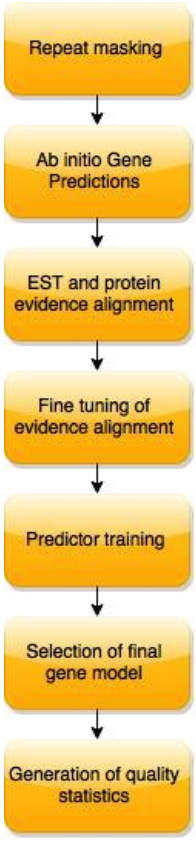


Figure 1: *Overview of the key steps MAKER [12] uses to produce genome annotations. Note that the steps are not as distinct as presented in the image. Many steps both overlap into each other as well as involve several different applications.*

Tophat-Cufflinks suite overview

The Tophat-Cufflinks [11] suite uses a workflow with several different steps that each have been encapsulated into individual programs (Figure 2). In short Tophat maps the sequence reads to a template genome. Cufflinks then assembles the reads into transcripts. Cuffmerge then produces a consensus transcriptome from the two (or more) assemblies.

Based on the consensus transcriptome as well as the individual assemblies Cuffdiff and Cuffnorm calculate deviations from the consensus. Finally visual results, such as graphs, are produced either through CummeRbund or R.

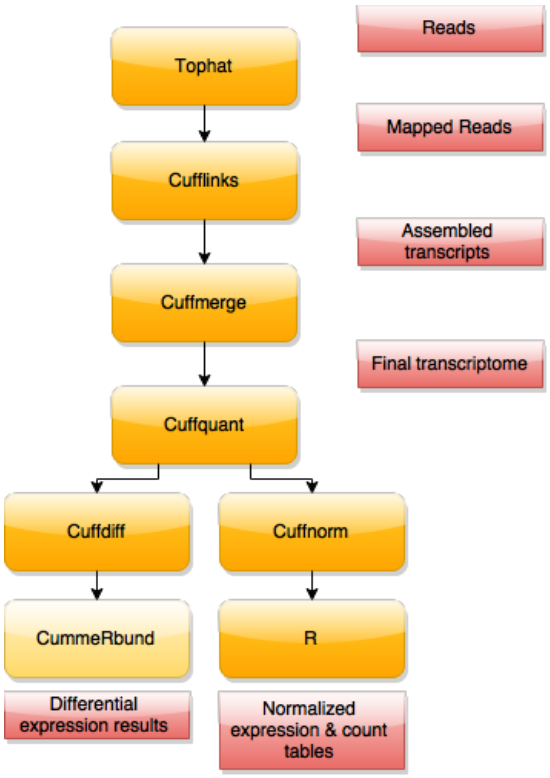


Figure 2: *Outline of how Tophat-Cufflinks [11] suite was used to generate data to support the hypothesis of differentially expressed genes.*

The pipeline for *S. digitata*

Assembly

As of writing there is no publically available assembled genome of *S. digitata*. The reference assembly we used for our analysis was created by our research member Arthur Perrad. By using multiple applications to assemble the reads as well sampling several different configurations he was able to produce several assemblies of adequate quality (Table 1). The assemblies were constructed without a reference by using QUAST [16]. MIRA [17] used an unpadded assembly on large contigs and Velvet [18] used k-mers of size 115. Other than that standard settings were used without any notable deviations. The best assemblies, primarily selected by their N50 values, are shown in Table 1. Out of the presented assemblies, we continued work solely on the Spades assembly.

Table 1: *Comparison of assemblies of S. digitata genome data.*

	# Contigs	Largest Contig	Total length	N50	Mismatches per 100kbp
Spades	41 945	190 831	110 339 552	10 397	0
Masurca	24 056	114 936	86 580 482	9 224	0
Velvet [18]	66 645	8 283	64 789 947	1 029	0
MIRA [17]	36 359	40 606	89 536 822	4 123	29.47

Annotation

Our selected genome assembly was annotated with the MAKER [12] pipeline, selected for its ease of use for annotation purposes. Two applications were omitted, tRNAscan-SE [19] and Snoscan [20]. This was done in part due to the fact that both applications had to be manually installed in addition to MAKER [12], thus increasing the difficulty in reproducing the pipeline. It was also done in part due to their limited applicability, as tRNAscan-SE [19] and Snoscan [20] only detected tRNA and snoRNA respectively.

Our MAKER [12] instance was ran on the UPPMAX [27] computational cluster. For this particular annotation pipeline a total of 11 prediction applications were used (Figure 3).

RepeatMasker [21]	Protein2Genome (built-in)	GeneMark-ES [22]	BLAST (n,x,tx,x) [23]
SNAP [24]	EST2Genome (built-in)	Augustus [25]	Exonerate [26]

Figure 3: *List of all prediction software MAKER [12] used for gene predictions in some way*

Four of the prediction applications required that a training set was chosen to model the respective application's predictor upon (Table 2). It was possible for us to form our own training set for *S. digitata* for the prediction applications, but was hindered due to limited public accessible data to train the predictors upon. The time and cost investments needed to generate adequate training were too high and we instead relied on training sets produced for the phylogenetically closest nematodes. This in turn meant we trained the predictors on training sets based on *B. malayi,* and in one instance for *Caenorhabditis elegans.*

Table 2: *Profiles used for gene prediction software where training upon generic underperformed (or did not work) compared to selecting a particular training set*

Name	Profile
Augustus [25]	*B. malayi*
GeneMark-ES [22]	*C. elegans*
RepeatMasker [21]	Te_proteins.fasta (manually chosen standard)
SNAP [24]	*B. malayi*

MAKER [12] also required EST and protein evidence to improve the prediction algorithms (Table 3). Since the available data for *S. digitata* was insufficient to make good predictions on its own we also included the entire superfamily for alternative EST evidence, as well as the entire invertebrate phyla for protein evidence.

Table 3: *Outside data used for prediction and their related sources*

Data type	Source
EST evidence	All *S.Digitata* evidence from the EST resource of NCBI (26 entries)
Alternative EST evidence	All filariodidea (superfamily) evidence from the EST resource of NCBI
Proteins	Uniprot database for invertebrates

Finally the results were visualized using the web Apollo [13] software on our research group's local server. Due to time constraints the visualization was not used to extensively search for any genes of interest in

this project. The results will however assist other researchers, more specialized in gene prediction, to apply the final layer of human curation needed to generate the final predictions.

Transcriptional differences in *P. polymyxa* A26

The laboratory analysis of *P. polymyxa* A26 was performed by Ignas Bunikis from Science for life laboratories (at Uppsala University) through the UPPNEX (UPPMAX Next Generation Sequencing Cluster & Storage) [27] platform. The analysis consisted of samples during two different conditions, stressed and neutral, for *P. polymyxa* A26. Each condition was divided into twelve Ion Xpress libraries. The data was single-ended with no mate-pairing or paired-ends. Prior to us receiving the data it was pruned of any sequencing primers such as barcodes and similar occurrences.

The gene expression data was inspected using the FastQC [28] tool which denounced it primarily for the unstable nucleotide ratios, overrepresentation of a subset of sequences and low phred-33 scores. In order to maintain as much sequencing data as possible with acceptable quality the libraries were trimmed to a lowest mean phred-33 score of 25 using PrinSeq [29]. On average this filtering retained one third of the original data for each library. After the trimming FastQC [28] still produced a multitude of warnings as FastQC [28] was designed with assembly and not gene expression in mind. Due to the circumstances the data post-filtering was considered of high enough quality.

For some of the analysis steps that the annotation pipeline performed, a reference genome was required. We used an unpublished *P. polymyxa* A26 genome that had both been sequenced and assembled in-house. The reference genome in turn used *P. polymyxa E681* (NCBI Reference Sequence: NC_014483.1) as a reference.

We annotated the reference genome using the automated RAST [30] pipeline, primarily due to its ease of use. RAST [30] did however introduce a few minor issues when set to work in conjunction with the rest of the pipeline. As such the RAST [30] output had to be manually reduced into unique entries and separate files had to be merged into a single one before continuing with the analysis. We automated this by programming a small script to solve the issue.

To call genes that had significant differences in expression levels we used the Tophat-Cufflinks [11] software suite. In brief the sequencing data was indexed using Bowtie [11], mapped for splice junctions using Tophat [11] and assembled and analyzed using different options in the Cufflinks [11] application. Some intermediate steps that were necessary were automated using perl scipts, which have been attached to the appendix of this paper.

The Cufflinks [11] application performed several key functions that are not readily apparent. First the libraries for both conditions (stressed and neutral) were assembled into two separate transcripts using the reference *P. polymyxa* A26 genome. The transcripts were then merged into a single consensus transcript using Cuffmerge [11]. The transcripts were compared to the consensus transcript to calculate if any significant deviations in gene expression levels were present, both between each other and individually against the consensus. Finally graphs were generated using the software R [31], some with the support of the cummeRbund [11] R package. Additional graphs were also generated using the graph generating script language Circos [32].

As Cufflinks searched for significance against identifiers in the consensus transcript, not the genes themselves, it was necessary to translate hits of significantly differentiating amounts into deviations on gene expression level. A perl script that automated the procedure was produced, and can be found in the ap-

pendix of this paper. Some of the genes were annotated as hypothetical genes by RAST [30]. In order to verify that the sequences were indeed hypothetical and not merely predicted as such by RAST [30], they were extracted and re-annotated by using BLASTx [23] (a tool which could almost be considered an industry standard at this point) to verify the results.

Bioinformatics for east Africa

The teaching segment of this project consisted of two segments. The first portion consisted of holding lectures and workshops in a week-long event alongside several other tutors, from Africa and USA at ILRI in Nairobi Kenya. Doctorates from all over eastern Africa were invited to this events as participants of this gathering. The second portion of the teaching segment consisted of a week-long workshop with three high-performing African doctorates who were flown cross-continent to participate in more advanced training at SLU in Sweden, Uppsala.

The event was focused on teaching the participants to solve their bioinformatical issues through simple means. In order to achieve this the participants learnt to use UNIX, the command line, NCBI software, GMOD annotation solutions and working with large computational clusters. Participants were also taught other skills for working in bioinformatics to various extents.

As a large portion of the teaching was done by other instructors than myself I will only be presenting the material I personally prepared and presented. As such some techniques and knowledge presented at the event will not be a part of this paper. All the material I produced has however been attached to the appendix of this paper and is more or less identical to the versions used, with the exception of some minor alterations that were done mid-teaching and has thus not made it into these copies.

Tuition at ILRI in Nairobi, Kenya

The tuition at ILRI was a collaborative project between several different academic institutions. Therefore the lectures and hands-on work we produced for this project only covered a few days of the week long event in east Africa. The teaching, as far as the project concerned, involved simple bioinformatical analysis by using predefined pipelines and posting the results on NCBI. To underline the real-life applications of generic bioinformatics pipelines usage of the MAKER [12] software suite was taught by taking examples from the annotation of *S. digitata.*

Due to time constraints, students were not tasked with annotating *S. digitata* but rather tasked with a custom simplified versions of MAKER's [12] tutorial. The necessary prerequisites for this tutorial was installed on ILRI's eBioKit [3]. In addition to this students were also tasked with posting their results on NCBI's web portal.

Advanced classes at SLU, Sweden

The teaching back at SLU consisted of more advanced bioinformatics training in one-on-one sessions with three top performing doctorates from the ILRI event. Students were taught how to install and run the UNIX operating system through a virtual machine; more advanced command line operations; how to install, use and customize an annotation pipeline for their needs and finally how to customize the MAKER [12] pipeline to solve their current research problems. In addition to this we also discussed practical solutions to bioinformatical problems that do not typically occur in a high-tech environment. One example of such was to minimize internet usage by copying as much information to their hard drives as possible as none of them had access to the internet on a regular basis.

Results

Bioinformatics for east Africa

The tutoring of students from developing countries produced very good results. Although we are unable to provide an empirically measurable metric of the quality of the teaching segments, we were left with the impression that the students had gained insight into the field which would help further their research. Questions relating to how they could incorporate bioinformatics into their research were quite common, and almost all of them could be resolved with simple modifications to the solutions presented.

In addition to this the level of difficulty seemed to be adequate. Despite students consistently asking questions not a single one was so stumped that they gave up or required constant hand-holding. The allotted time was enough for over 90% of the students to finish the workshops they were assigned during the event.

Annotation of the *S. digitata* genome

All annotation material produced by the MAKER [12] pipeline was saved on SLU's local computational cluster planetsmasher and manually reviewed as a means of quality control. After the project the results were visualized in Web Apollo [13] (Figure 4) by Jonas Söderberg in order to allow other scientists to more easily assess potential gene homologies to *W. bancrofti*.

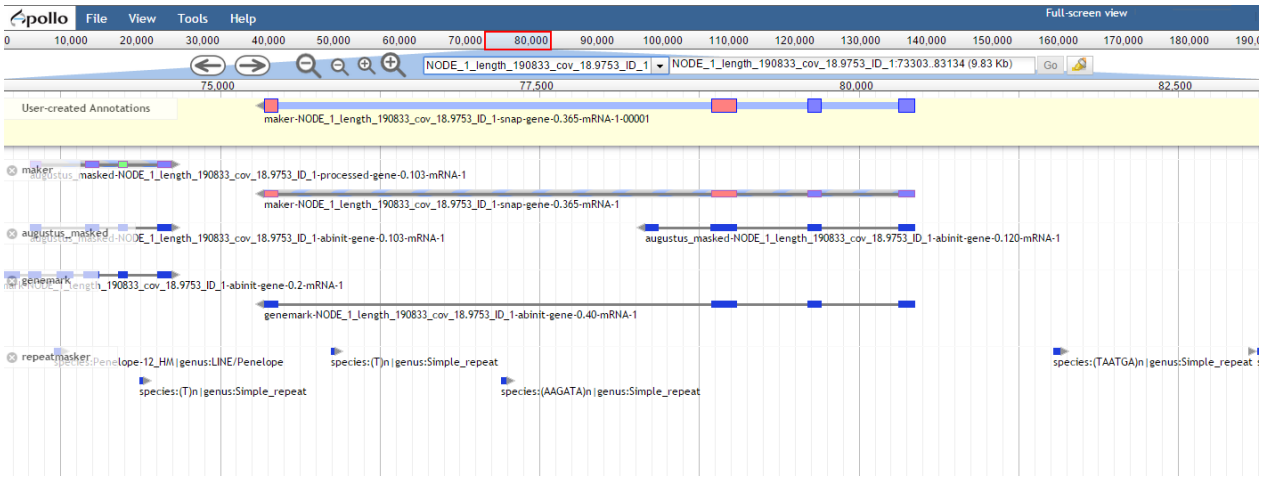


Figure 4: *Screen capture of web Apollo [13] loaded with the S. digitata data. The picture depicts gene evidence for a small genomic region with predictions from RepeatMasker [21], GeneMark-E [22], Augustus [25] and compound MAKER [12] predictors.*

Differentially expressed genes in *P. polymyxa* A26

The algorithm Cufflinks [11] used for determining significant deviation (a comparison of p-value against the false detection rate after Benjamini-Hochberg correction [33]) deemed only eight genes as significantly deviating in gene expression levels. Almost half of these were exclusively annotated as hypothetical proteins.

Cufflinks [11] measures whether a given entry is considered significant or not by combining several methods of value pair comparison to reach a binary decision. In brief the p-value for the sample is calculated based on a Student's t-test. Q-values are then generated by simply correcting the p-values for false detection rate. After applying a Benjamini-Hochberge correction the q-values are then compared to the p-

values. Whether the difference between the two exceeds a predetermined threshold value dictates whether the deviation in gene expression levels is considered significant or not.

As our study only yielded eight candidates it we assumed that the conditions were too stringent. As such we also included all hits where the p-value exceeded 5 percent. This more lenient approach produced significance for 106 named genes and 10 hypothetical ones, including the ones resulting in Cufflinks [11] more stringent criteria. Accounting for entries that resolved back to the same gene expression**,** 98 unique genes were found to have significantly deviating gene expression levels. This was a far more reasonable result in comparison to the output to other studies [34][35]. We do however note the significant loss of robustness as compared to Cufflinks [11] internal method. The full list of results has been attached in Table 7 of the appendix.

Approximately ten percent of the differentially expressed genes with a p-value under 5% could only be annotated as hypothetical proteins by RAST [30]. We extracted these sequences and re-annotated them using BLASTx [23] against the non-redundant protein sequence database. Out of the 14 hits presented as hypothetical genes, four could be resolved (Table 4). Out of the four entries, two were unique and also mapped to *P. Polymyxa* from prior studies.

Table 4: *Differentially expressed genes with a p-value under 5% annotated as hypothetical by RAST [30]. The significant field refers to whether the hit was significant or not according to Cufflinks [11] internal threshold value.*

Function	Significant	E-val	Mapped to P. Polymyxa
Sugar ABC transporter substrate-binding protein	yes	0	yes
Sugar ABC transporter substrate-binding protein	no	7E-28	yes
Acyl Carrier Protein	no	2E-43	yes
Chromosome Partitioning protein ParA	yes	1E-07	no

Data validation of *P. polymyxa* A26 transcripts

In order to validate that the sequence data was representative of the transcriptome of *P. polymyxa* A26 the gene expression data was used to assemble a transcriptome. The purpose of this was not to create a fully functional transcriptome, but rather verify that a representative portion of the genome had been transcribed.

The gene expression data was initially visually inspected using the FastQC [28] tool which denounced it primarily for the unstable nucleotide ratios, overrepresentation of a subset of sequences and low phred-33 scores.

Table 5*: Summary of sequence trimming*

Deduplication	Exact, 5', 3', exact compliment
Left-hand trimming	10 bases
Right-hand trimming	Quality score above 24
Trimmed length	230 bases
Retained data	11.3 GB (17.8%)

In order to maintain as much sequencing data as possible with acceptable quality the libraries were trimmed using several iterations of PrinSeq [29] (Table 5). The overrepresentation was handled by removing exact duplicates, 5' duplicates, 3' duplicates and reverse compliment exact duplicates. The sequences were also trimmed from the right-hand side to a phred-33 score of 24. All libraries were then trimmed from the left-hand side to remove low-quality sequence ends. After a thorough secondary visual examination it was concluded that the first ten bases of all sequences had to be cut. Finally to resolve the unstable nucleotide ratios towards the 5' ends; All sequences, with the exception of the sequences found in the last five batches for the first condition of the organism, were trimmed down to a total length of 230 nucleotides.

Following these pruning steps all sequences scored over 20 points of base sequence quality of phred-33 score (Figure 5). Out of the 63.6 GB of sequencing data, roughly 11.3 GB (or 17.8%) were retained. FastQC [28] still warned about K-mer overrepresentation. Considering that the software is typically used for genome and not transcriptome analysis the warning was ignored.

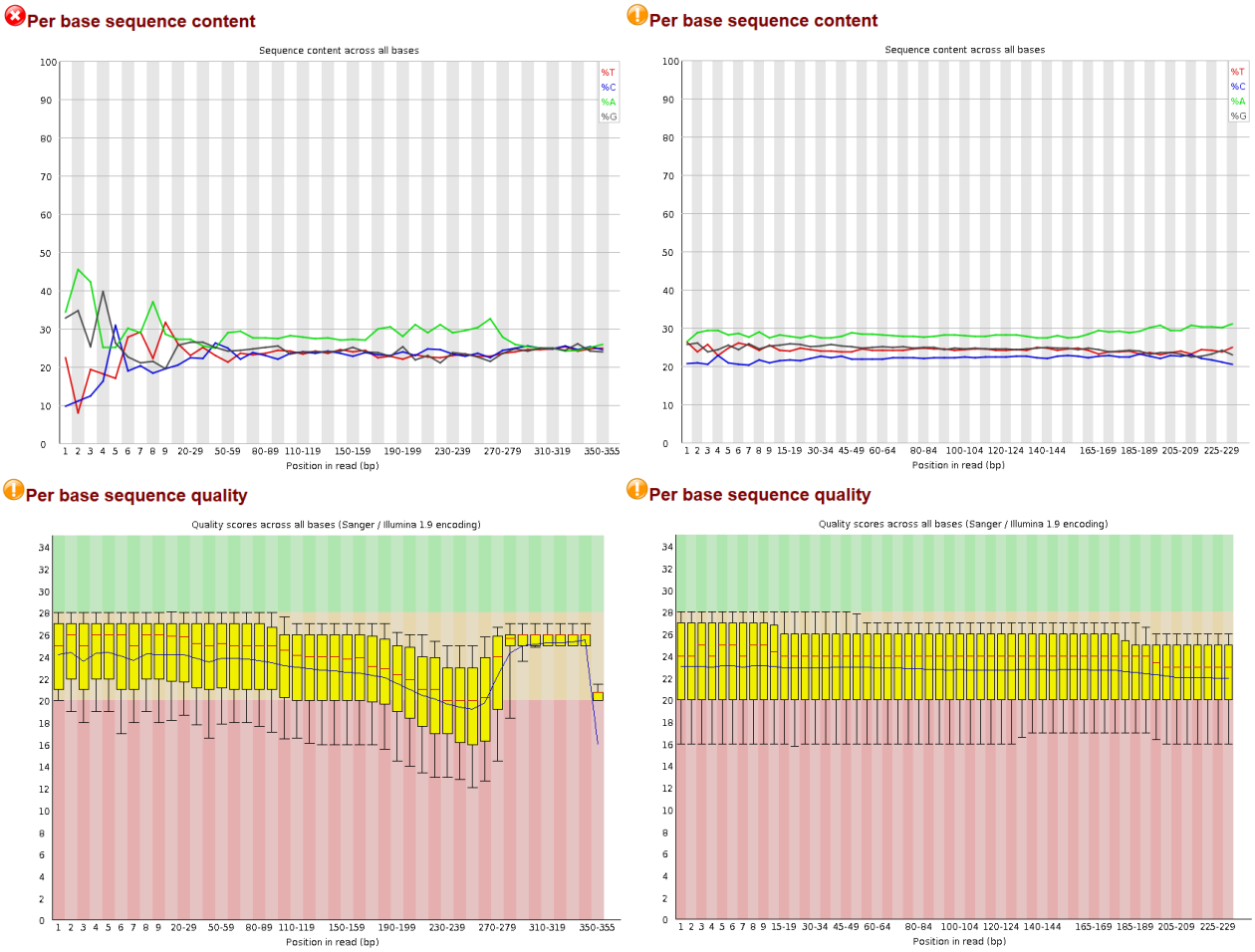


Figure 5: *Graphical comparison of the libraries through* FastQC[28] *after deduplication and trimming. The images show the concatenated library for sample 2, both before (left) and after (right) the filtering. The upper images depict the nucleotide ratios, which are expected to be even. The lower images depict the average phred-33 score for each base pair of sequences.*

The transcriptome data was then concatenated and assembled with several different assembly applications. As our initial assembly produced notably poor results we used multiple assemblers to validate that the interaction between the data and the particular software was the source of the error, rather than the data itself.

The sequences were assembled with MIRA [17], Trinity [36], Trans-ABySS [37], SOAPdenovo [38] and Oases [39] using their respective default settings but with several approaches. For Oases [39] we in addition to a typical run also merged the results of several different k-mer runs. For Trinity [36] we assembled both with and without its genome guided function.

The quality of the assemblies was determined using QUAST [16]. Arguably the non-guided Trinity [36] assembly provided the best results. The best assembly was determined by factoring in several variables such as N50, covered genome fraction and duplication ratio. The gene expression data for this particular assembly represented over three thirds of the genome and as such the sequence data was deemed fit for further analysis.

The pooled transcriptome libraries were compared against the previously mentioned in-house reference assembly of P. polymyxa A26 (Table 6). The non-guided Trinity [36] assembly provided the best results by merit of having the highest N50, highest genome fraction and low duplication ratio. The Trans-abyss assembly was a close second.

Table 6: *Statistics for the assemblies generated by pooling transcriptome libraries*

Statistics without reference	Oases23-29merge	oases23	oases25	oases27	oases29	SOAPdenovo	transAbyss	trinityGenomeGuided	trinityNoGenomeGuide
# contigs	16 825	6580	6351	6736	6398	1541	5661	190	3726
Largest contig	73 588	45 247	68 708	54 682	73 435	3625	9924	4961	22 270
Total length	21 962 391	8 461 760	8 298 322	8 816 207	8 000 244	1 146 818	5 940 824	158 245	6 565 304
N50	1562	1581	1559	1575	1494	719	1148	789	2400
Misassemblies									
# misassemblies	829	165	181	183	270	0	103	8	104
Misassembled contigs length	1 638 187	289 394	453 660	382 423	591 167	0	96 313	7890	337 798
Mismatches									
# mismatches per 100 kbp	1351.15	1243.91	1245.43	1255.57	1256.64	641.82	1261.81	41.21	1263.46
# indels per 100 kbp	473.29	453.88	433.53	432.43	406.62	101.83	117.92	217.7	143.71
# N's per 100 kbp	0	198.34	211.58	183.78	141.85	0	0	0	0
Genome statistics									
Genome fraction (%)	70.304	43.367	42.965	43.572	41.429	14.569	74.192	2.515	83.918
Duplication ratio	4.299	2.676	2.644	2.789	2.692	1.003	1.176	1.072	1.159
NGA50	3103	1860	1767	1919	1632	-	1023	-	2340

Graphic assessment of differentially expressed genes in *P. polymyxa* A26

In order to more easily visualize the validity of the suggested significantly differently expressed genes; R [31] with the cummeRbund [11] package was used to graph the distinction between the gene levels deemed significant and non-significant. A heat map of the results (Figure 6) showed that there is a distinction in gene levels by at least a factor 10 for those deemed significant.

Figure 6: *Heat map of differentially expressed genes with a p-value under 5%. A more intense orange signifies a higher level of expression. Q1 and Q2 refers to condition one and two respectively.*

To further elaborate on this point a volcano plot was generated to show the distinction between the differentiating gene levels deemed significant by Cufflinks [11] internal threshold and a p-value of 5 percent; compared to all the differentiating gene levels of the analysis (Figure 7). In both cases the images show that both thresholds produce similar results. The threshold of a 5 percent p-value merely alters the amount of significantly differentiating gene levels included, and does not filter out entries that one would otherwise expect to still be retained.

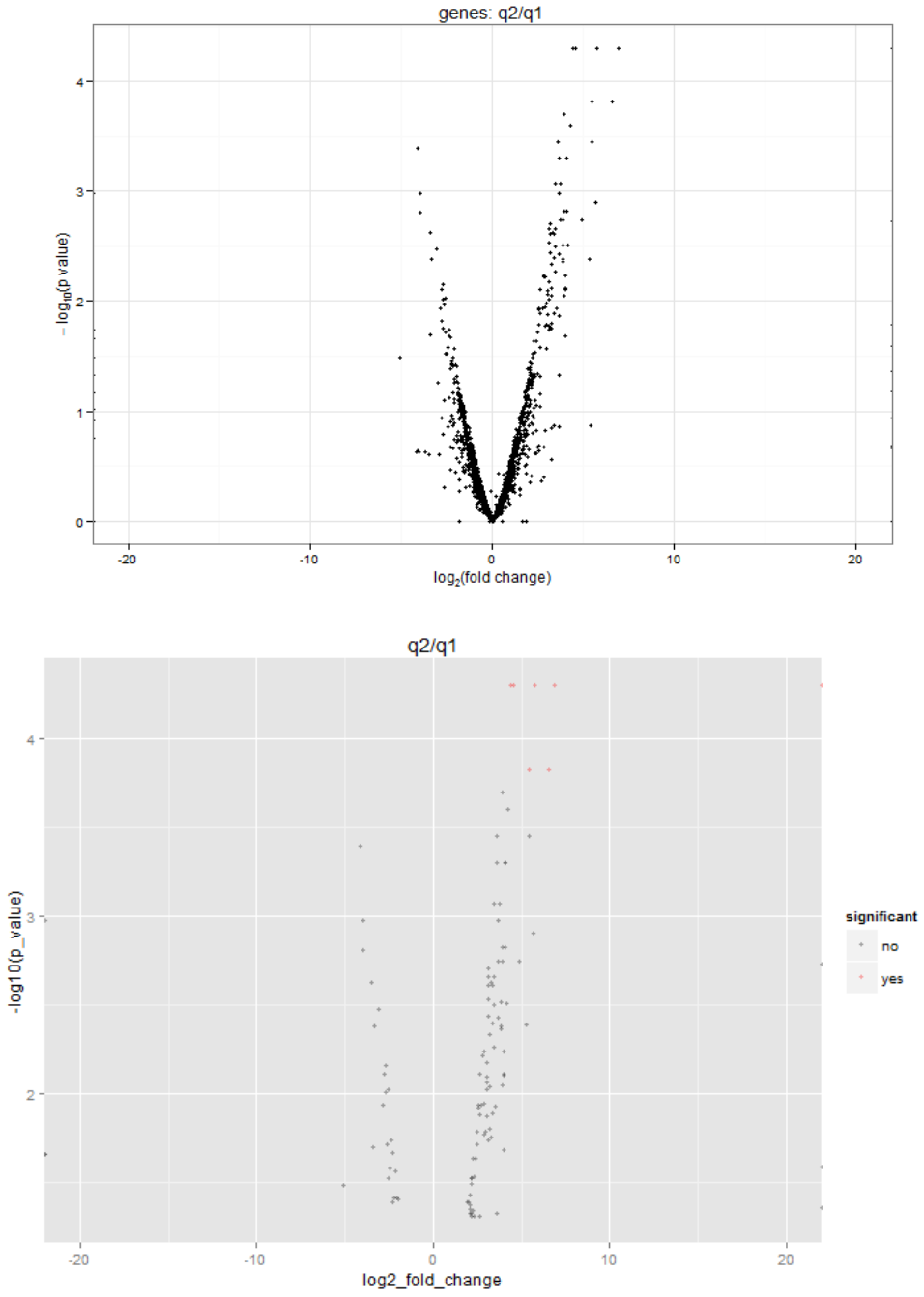


Figure 7: *Volcano plots of all genes (top) and those differentially expressed with a p-value under 5% (bottom). Entries deemed significant by Cufflinks internal algorithm are highlighted in orange.*

Differentially expressed genes in relation to the transcriptome in *P. polymyxa* A26

To gain a quick overview of what sections of the transcriptome had been differentially expressed under the differing conditions we base called all the hits that were deemed significant by our extended threshold criteria (a p-value exceeding 5 percent) back to the genome we used as a basis for our transcriptome construction. By using Circos [32] to plot the results, it became clear that almost all of the differentiating genes were located between 2.36M and 4.98M (Figure 8).

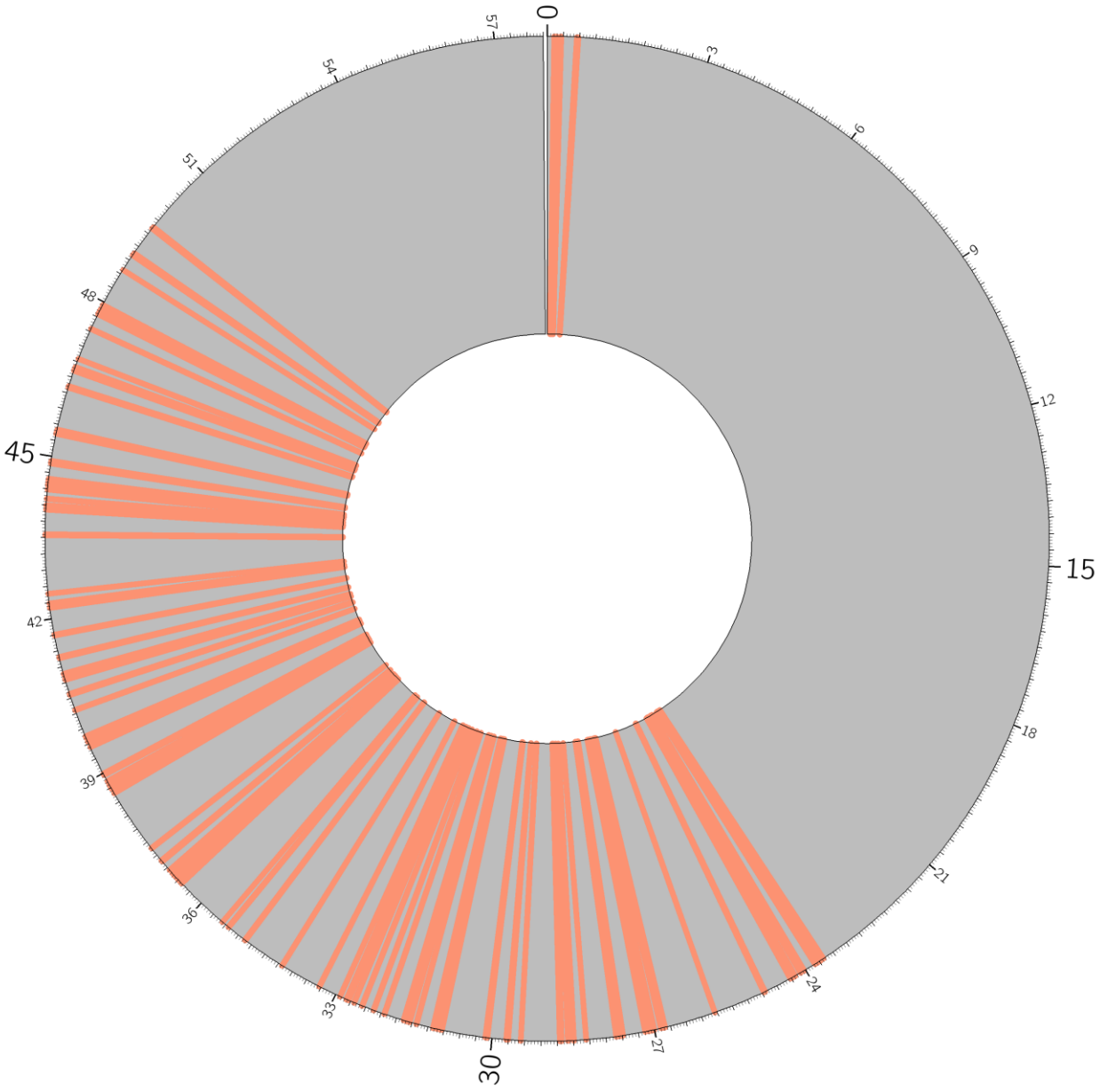


Figure 8: *Ideogram in the scale of 100 000 base pairs, showing the approximate positions of all differentially expressed genes with a p-value under 5%. Length of individual transcripts have been greatly exaggerated for this visual representation.*

Discussion

Pipelines as a SOP for bioinformatics in Africa

This project presented two pipelines general enough to be used by novice bioinformaticians for typical bioinformatical analysis. One pipeline related to expressional differences in the transcriptome, and the other one related to the annotation of a genome. Although some minor scripting had to be done and some settings had to be altered, the solutions required almost no manual set-up.

As part of the teaching segment of the project, a simplified version of genome annotation through MAKER [(12)] was used during the workshop. East African students with virtually no prior bioinformatical knowledge got through it with very few hiccups and actively experimented with how it could it be applied to their research. I believe that both pipelines could be used by eager academics in other developing countries with very minor alterations to account for their research.

As I have a background in assisting first year students with programming at Uppsala University, Sweden, I expected a similar level of motivation and expertise from the east African doctorates. The short term progress did however blow me out of the water and personally showed me how much I underestimated their desire to learn bioinformatics. From a starting point where several students were unable to even remember their own passwords, we left with several students able to fluently use the command line in UNIX and even run typical bioinformatical software with only a few days of practice. Given enough resources I honestly believe many of them would be able to rise up to western standards.

Possible extensions

This project consisted of four different sub-project, each which could be further improved. In no particular order these relate to the *E. coracana genome annotation project*, the *S. digitata* genome annotation project, the *P. polymyxa* A26 transcript differentiation and the east African bioinformatical resources.

Improvements to the transcriptional differences in *P. polymyxa* A26

For *P. polymyxa* A26 we used one typical pipeline for generating the results. Naturally one could run several fundamentally different pipelines and comparatively analyze them. The results could also be further verified by laboratory analysis. Given the relatively limited scope of the sub-project, it does however feel like an adequate amount of work was put into it.

Improvements to the annotation of *E. coracana & S. digitata*

The annotations for *S. digitata* are as of writing internally available for our research group and viewable through the web Apollo [(13)] browser. The only remaining steps is to allocate the resources required to sift through the data and curate it as well assessing potential homologies to *W. bancrofti.* The annotations themselves could also be further improved by using the curated annotations to train and re-run MAKER [(12)] to potentially find other genes that currently are not predicted.

Based on the findings of annotating *S. digitata* is it very reasonable to believe that a nearly identical approach can be used to generate annotation data for *E. coracana.* As with *S. digitata* the suggested annotations will have to be manually curated before publication. The biggest difference between the two genomes is their respective size. As only internal pre-assembly data of *E. coracana's* genome is currently available no definitive answer can be given as to how big the genome actually is. Suffice it to say, it will

require more than just a single bioinformatician to review the annotations in a timely matter. One possible way to resolve this is to involve interested academics from east Africa for the project as the region hosts good biotechnical resources.

Improvements to the Bioinformatics in east Africa project

East Africa shows a lot of promise as a bioinformatical resource. It is currently in a very rough state, mainly due to a few key factors, but were they to improve I feel there is a great potential ready to be utilized. In the future east Africa could not only be used as a cheap way to produce bioinformatical results, but also as a way for east African nations to gain bioinformatical independence from other nations for their own research.

In regards to the key factors that could see improvement to better the bioinformatical science in East Africa; I believe the biggest hurdle this community has when attempting to excel at bioinformatics is not in the tuition itself but rather the lack of proper infrastructure. To name a few:

- The power grid is very unstable, IT work is thus limited to facilities with a generator
- The internet access for many regions is practically non-existent and as such commonplace features such as googling answers, cheap conference calls and emergent cloud solutions are unavailable
- Relatively few bioinformaticians reside in the area. As such *asking your local expert* is almost never a possibility.

The unstable power grid is manageable, and with eBioKits [(3)] provided to many institutions the reliance on internet has been drastically reduced. What the region primarily needs is more local bioinformatical experts to help with IT set-up, administration and support in bioinformatical issues.

Evaluation the effectiveness of these actions would most likely be surprisingly simple. In this project we introduced the usage of what we believe to be the easiest way to process transcriptome data and annotate a genome. Typical bioinformatical tasks such as assembling or annotating a genome is usually only difficult in regards to the structure of the genome and the tools used. As such one would expect academics of the region to be more proficient in handling more complex forms of analysis as the underlying infrastructure improves.

Acknowledgements

I would like to thank my professor and supervisor Erik Bongcam-Rudloff for the possibility to work on the project. I would also like to thank Arthur Perrad and Jonas Söderberg for their work on critical parts of the *S. digitata* segment of the project.

I would also like to thank the Department of Animal Breeding and Genetics at SLU, ILRI and UPPMAX for their financial- , cooperative and computational support respectively.

Finally I would like to thank my girlfriend Malin for her support as I struggled to finish this project. As six months became twelve I became increasingly uncertain of my ability to finish it at all. Thank you for all the emotional support I have been given.

References

[1] Watts M. *Entitlements or empowerment? Famine and starvation in Africa. Review of African Political Economy* (1991) 9-26 DOI: 10.1080/03056249108703903

[2] *The Economist. Better dead than GM-fed? The Economist Newspaper Limited 2016 (2002-09-22)*. URL: http://www.economist.com/node/1337197

[3] Bongcam-Rudloff E. *eBioKit Project. Erik Bongcam-Rudloff Group* (2013) URL: http://www.ebioinformatics.org/

[4] United Nations Statistics Division. *Composition of macro geographical (continental) regions, geographical sub-regions, and selected economic and other groupings. United Nations* (2014) URL: http://millenniumindicators.un.org/unsd/methods/m49/m49regin.htm

[5] Corbett J. *Famine and household coping strategies. World Development* (1988) 16:9:1099-1112. Doi: 10.1016/0305-750X(88)90112-X

[6] Ghannoum O., Caemmerer S. and Conroy J. P. *The effect of drought on plant water use efficiency of nine NAD–ME and nine NADP–ME Australian C4 grasses. Functional Plant Biology* (2002) 29(11) 1337 – 1348 Doi: 10.1071/FP02056

[7] Food and Agriculture Organization of the United Nations. *Top Production – World. FAO 2013 (2013)* URL: http://faostat.fao.org/site/339/default.aspx

[8] Addiss DG, Eberhard ML, Lammie PJ, McNeeley MB, Lee SH, McNeeley DF, Spencer HC. *Comparative efficacy of clearing-dose and single high-dose ivermectin and diethylcarbamazine against Wuchereria bancrofti microfilaremia. The American Journal of Tropical Medicine and* Hygiene (1993) 48(2):178-185 PMID: 8447520

[9] Ottesen EA, Weller PF, Heck L. *Specific cellular immune unresponsiveness in human filariasis. Immunology*. 1977;33(3):413–421. PMCID: PMC1445652

[10] Michael E, Bundy DA, Grenfell BT. *Re-assessing the global prevalence and distribution of lymphatic filariasis. Parasitology.* 1996 Apr;112(Pt 4):409–428. PMID: 8935952

[11] Trapnell C, Roberts A, Goff L, Pertea G, Kim D, Kelley DR et al. *Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. Nature protocols* (2012) 562–578. PMID: 22383036.

[12] Holt C. & Yandell M. *MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. BMC Bioinformatics (*2011) 12:491. Doi: 10.1186/1471-2105-12-491

[13] Lee E, Helt GA, Reese JT, Munoz-Torres MC, Childers CP, Buels RM, Stein L, Holmes IH, Elsik CG, Lewis SE. *Web Apollo: a web-based genomic annotation editing platform. Genome Biology* (2013) 14:R93. doi:10.1186/gb-2013-14-8-r93

[14] Yandell Lab. *MAKER Overview. Mark Yandell* (2007-2011) URL: http://www.yandell-lab.org/software/maker.html

[15] Trapnell C. *Cufflinks: Transcriptome assembly and differential expression analysis for RNA-Seq. Cole Trapnell* (2016) URL: http://cole-trapnell-lab.github.io/cufflinks/

[16] Gurevich A, Saveliev V, Vyahhi N & Tesler G, *QUAST: quality assessment tool for genome assemblies. Bioinformatics* (2013) 29 (8): 1072-1075. Doi: 10.1093/bioinformatics/btt086

[17] Chevreux, B., Wetter, T. and Suhai, S. *Genome Sequence Assembly Using Trace Signals and Additional Sequence Information. Computer Science and Biology: Proceedings of the German Conference on Bioinformatics* (1999) 99:45-56.

[18] Zerbino DR, Birney E. *Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res* (2008) 18(5):821-9. PMID: 18349386

[19] Lowe T. M. & Eddy S. R. *tRNAscan-SE: A Program for Improved Detection of Transfer RNA Genes in Genomic Sequence. Nucleic Acids Res.* (1997) 25 (5): 0955-964. Doi: 10.1093/nar/25.5.0955

[20] Lowe T. M. & Eddy S. R. *A computational screen for methylation guide snoRNAs in yeast. Science* (1999) 283(5405):1168-71. PMID: 10024243

[21] Smit AFA, Hubley R & Green P. *RepeatMasker Open-4.0. Institute for Systems Biology* (2013-2015) URL: http://www.repeatmasker.org

[22] Ter-Hovhannisyan V, Lomsadze A, Chernoff YO, Borodovsky M. *Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training. Genome Research* (2008) (12):1979-90. PMID: 18757608

[23] Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. *Basic local alignment search tool. J. Mol. Biol.* (1990) 215:403-410. PMID: 2231712

[24] Leskovec J. and Sosi R. *SNAP: A general purpose network analysis and graph mining library in C++. Stanford University* (2014) URL: http://snap.stanford.edu/snap

[25] Stanke M. and Waack S. *Gene prediction with a hidden Markov model and a new intron submodel. Bioinformatics* (2003) 19 (suppl 2): ii215-ii225. Doi: 10.1093/bioinformatics/btg1080

[26] Slater G. S. C. & Birney E. *Automated generation of heuristics for biological sequence comparison. BMC Bioinformatics* (2005) 6:31 Doi: 10.1186/1471-2105-6-31

[27] Lampa S, Dahlö M, Olason PI, Hagberg J, Spjuth O. *Lessons learned from implementing a national infrastructure in Sweden for storage and analysis of next-generation sequencing data. Gigascience* (2013) 2(1):9. Doi: 10.1186/2047-217X-2-9.

[28] Patel R. K., Mukesh J. *NGS QC Toolkit: A Toolkit for Quality Control of Next Generation Sequencing Data. PlosOne* (2012) 7(2). PMCID: PMC3270013

[29] Schmieder R and Edwards R: *Quality control and preprocessing of metagenomic datasets. Bioinformatics* (2011) 27:863-864. PMID: 21278185

[30] Aziz RK, Bartels D, Best AA, DeJongh M, Disz T, Edwards RA, Formsma K, Gerdes S, Glass EM, Kubal M, Meyer F, Olsen GJ, Olson R, Osterman AL, Overbeek RA, McNeil LK, Paarmann D, Paczian T, Parrello B, Pusch GD, Reich C, Stevens R, Vassieva O, Vonstein V, Wilke A, Zagnitko O.

*The RAST Server: rapid annotations using subsystems technology. BMC Genomics (*2008) 9:75. PMID: 18261238

(31) R Core Team. *R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing* (2015) URL: http://www.R-project.org

(32) Krzywinski, M. et al. *Circos: an Information Aesthetic for Comparative Genomics. Genome Research* (2009) 19:1639-1645. Doi: 10.1101/gr.092759.109

(33) Benjamini Y. & Hochberg Y. *Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society, Series B* (1995) 57 (1): 289–300. MR 1325392

(34) Griffith M. et al. *Alternative expression analysis by RNA sequencing. Nature Methods* (2010) 7:843–847 Doi: 10.1038/nmeth.1503

(35) Lin M. et al. *RNA-Seq of Human Neurons Derived from iPS Cells Reveals Candidate Long Non-Coding RNAs Involved in Neurogenesis and Neuropsychiatric Disorders. PlosOne* (2011) DOI: 10.1371/journal.pone.0023356

(36) Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, Adiconis X, Fan L, Raychowdhury R, Zeng Q, Chen Z, Mauceli E, Hacohen N, Gnirke A, Rhind N, di Palma F, Birren BW, Nusbaum C, Lindblad-Toh K, Friedman N, Regev A. *Full-length transcriptome assembly from RNA-seq data without a reference genome. Nature Biotechnology. (*2011) 29(7):644-52. PMID: 21572440.

(37) Robertson G., Schein J., Chiu R., Corbett R., Field M., Jackman S. D., Mungall K., Lee S., Okada H. M., Qian J. Q., Griffith M., Raymond A., Thiessen N., Cezard T., Butterfield Y. S., Newsome R., Chan S. K., She R., Varhol R., Kamoh B., Prabhu A-L., Tam A., Zhao Y., Moore R. A., Hirst M., Marra M. A., Jones S. J. M., Hoodless P. A. & Birol I. *De novo assembly and analysis of RNA-seq data. Nature Methods* (2010) 7, 909–912. Doi: 10.1038/nmeth.1517

(38) Luo R. et al. *SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. GigaScience* (2012) 1:18. Doi: 10.1186/2047-217X-1-18

(39) Schulz M. H., Zerbino D. R., Vingron M. & Birney E. *Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. Bioinformatics* (2012) 28(8): 1086–1092. PMCID: PMC3324515

Appendix

Suggested differentially expressed genes in *P. polymyxa* A26

Table 7: *Differentially expressed genes with a p-value under 5% that were annotated through RAST [30]. Duplicate entries exist as multiple sequences coded for the same item. The significant field refers to whether the hit was significant or not according to Cufflinks [11] predefined threshold value. Entries are sorted by p-value.*

Function	P-value	Significant
5S RNA	5.00E-05	yes
Stage II sporulation serine phosphatase for sigma-F activation	5.00E-05	yes
Probable ABC transporter permease protein ytcP	5.00E-05	yes
NifU-like domain protein	0.00015	yes
Cellulase	0.00025	no
Xylose ABC transporter2C permease component	0.00035	no
Xylose ABC transporter2C permease component	0.00035	no
acyltransferase family protein	0.0004	no
Cellulase	0.0005	no
Iron-sulfur cluster assembly protein SufD	0.0005	no
Glycerol kinase	0.00085	no
protein of unknown function DUF990	0.00085	no
Polysaccharide ABC transporter substrate-binding protein	0.00105	no
Serine/threonine protein kinase	0.00105	no
transcriptional regulator2C Crp/Fnr family	0.00105	no
putative cellulose 12C4-beta-cellobiosidase	0.00125	no
Endo-12C4-beta-xylanase A precursor	0.0015	no
Cellulase	0.0015	no
Alpha-amylase	0.0018	no
Cellobiose phosphorylase	0.0018	no
Iron-sulfur cluster assembly protein SufB	0.0018	no
Xylose ABC transporter2C permease component	0.00195	no
Altronate oxidoreductase	0.0022	no
Altronate dehydratase	0.00245	no
DNA-binding protein HBsu	0.00235	no
RNA polymerase sporulation specific sigma factor SigG	0.0022	no
2-keto-3-deoxygluconate permease	0.00245	no
Small Subunit Ribosomal RNA%3B ssuRNA%3B SSU rRNA	0.0031	no
Hydroxymethylpyrimidine ABC transporter2C transmembrane component	0.00295	no
Diaminopimelate decarboxylase	0.00315	no
Large Subunit Ribosomal RNA%3B lsuRNA%3B LSU rRNA	0.00305	no
Endo-beta-12C3-12C4 glucanase	0.00365	no
Sporulation sigma-E factor processing peptidase	0.0037	no

RNA polymerase sporulation specific sigma factor SigE	0.004	no
Transcriptional regulator2C MarR family	0.00415	no
Aspartyl-tRNA	0.00415	no
Glycerol uptake facilitator protein	0.0043	no
Myo-inositol 2-dehydrogenase	0.0046	no
tRNA-Trp-CCA	0.0054	no
Endoglucanase E1 precursor	0.0058	no
Xylose ABC transporter2C substrate-binding component	0.00575	no
ABC-type polysaccharide transport system2C permease component	0.00605	no
Acetyl-coenzyme A synthetase	0.00665	no
amino acid permease family protein	0.00695	no
rhamnogalacturonan acetylesterase	0.00775	no
ABC transporter ATP-binding protein	0.0078	no
AstB/chuR/nirj-related protein	0.00765	no
glycosyl transferase2C family 2	0.0077	no
YTEU	0.00805	no
Possible D-alanyl carrier protein2C acyl carrier protein family	0.00865	no
Similar to C-ter. fragment of UDP-glucuronosyltransferases2C YpfP B.subtilis related	0.00885	no
Hydrolase2C haloacid dehalogenase-like family	0.009	no
Anthranilate phosphoribosyltransferase	0.00935	no
putative unsaturated glucuronyl hydrolase	0.0095	no
Tryptophan synthase alpha chain	0.0097	no
Rhamnulokinase	0.0115	no
surfactin production and competence	0.01155	no
Uncharacterized protein2C 4-oxalocrotonate tautomerase homolog	0.0113	no
Sulfate permease	0.01155	no
Succinate dehydrogenase flavoprotein subunit	0.0117	no
Polysaccharide ABC transporter substrate-binding protein	0.0119	no
Succinate dehydrogenase iron-sulfur protein	0.0128	no
Polygalacturonase	0.013	no
Glycosyltransferase2C MGT family	0.01325	no
L-rhamnose isomerase	0.0163	no
Beta-xylosidase	0.01615	no
tRNA-Met-CAT	0.0157	no
lipoprotein	0.0168	no
sugar ABC transporter permease	0.01745	no
glycosyl transferase2C family 2	0.0181	no
Acetylornithine aminotransferase	0.0183	no
Malonyl CoA-acyl carrier protein transacylase	0.01935	no
Peptidyl-prolyl cis-trans isomerase	0.01935	no
amidohydrolase	0.01995	no

N-Acetyl-D-glucosamine ABC transport system2C permease protein 1	0.0208	no
Beta-galactosidase	0.02185	no
2-dehydropantoate 2-reductase	0.02125	no
Chaperone protein DnaJ	0.02185	no
Bacitracin synthetase 3	0.02305	no
Alfa-L-rhamnosidase	0.02315	no
Mobile element protein	0.0255	no
Formate hydrogenlyase subunit 7	0.0262	no
Cytochrome d ubiquinol oxidase subunit I	0.02695	no
arabinogalactan endo-12C4-beta-galactosidase	0.029	no
MazG nucleotide pyrophosphohydrolase	0.02975	no
DNA-binding response regulator2C AraC family	0.02995	no
Xylose ABC transporter2C permease component	0.0296	no
Putative exported protein precursor	0.03275	no
Xylose isomerase	0.0322	no
Chromosome initiation inhibitor	0.0369	no
Putative FMN hydrolase	0.0385	no
NADP-specific glutamate dehydrogenase	0.0383	no
Multidrug resistance protein [function not yet clear]	0.0394	no
DUF1541 domain-containing protein	0.04085	no
Multiple sugar ABC transporter2C membrane-spanning permease protein MsmG	0.0408	no
transmembrane lipoprotein	0.04045	no
Beta-glucosidase	0.04235	no
Two-component response regulator yesN	0.04675	no
FMN reductase	0.04525	no
Alcohol dehydrogenase	0.04685	no
N-Acetyl-D-glucosamine ABC transport system2C sugar-binding protein	0.0442	no
tRNA-Gly-TCC	0.0438	no
Transcriptional regulator2C MarR family	0.0469	no
Signal peptidase-like protein	0.0483	no
Glucose/mannose:H+ symporter GlcP	0.04875	no
maltose/maltodextrin transport permease	0.04905	no

Pipeline for running the Tophat-Cufflinks [11] suite through UPPMAX

```
#!/bin/bash

#SBATCH -A b2014195
#SBATCH -J tophatA26
#SBATCH -p node -n 12
#SBATCH -t 30:00:00

module load bioinfo-tools cufflinks/2.2.1 tophat/2.0.4 bowtie2/2.2.3 samtools


##1)Set index

export BOWTIE2_INDEXES=/proj/b2014195/nobackup/data/A26/indexedData


###2)Build index

#bowtie2-build -f /proj/b2014195/2014Bong/data/A26/A26.fa A26


##3) Running tophat2 withOUT A26 rast GTF reference:

#mkdir lib1
tophat2 -p 12 -o ./lib1TH /proj/b2014195/nobackup/data/A26/indexedData/A26
/proj/b2014195/nobackup/data/A26/cut/lib1/lib1cutpool.fastq

#mkdir lib2
tophat2 -p 12 -o ./lib2TH /proj/b2014195/nobackup/data/A26/indexedData/A26
/proj/b2014195/nobackup/data/A26/cut/lib2/lib2cutpool.fastq


##Convert tophat output to sam

cd lib1TH/
samtools view accepted_hits.bam > ./accepted_hits.sam
cd ..
cd lib2TH/
samtools view accepted_hits.bam > accepted_hits.sam
cd ..

##Run cufflinks with GTF reference

cufflinks -G /proj/b2014195/nobackup/data/A26/indexedData/A26.gtf -o
./cufflinkslib1 ./lib1TH/accepted_hits.sam

cufflinks -G /proj/b2014195/nobackup/data/A26/indexedData/A26.gtf -o
./cufflinkslib2 ./lib2TH/accepted_hits.sam
```

```
##Cuffmerge the two cufflinks entries

echo "/proj/b2014195/scripts/cufflinkslib1/transcripts.gtf" > cuffmerge.list
echo "/proj/b2014195/scripts/cufflinkslib2/transcripts.gtf" >> cuffmerge.list
cuffmerge -g /proj/b2014195/nobackup/data/A26/indexedData/A26.gtf -s
/proj/b2014195/nobackup/data/A26/indexedData/A26.fa -o ./cuffmerge cuff-
merge.list


##Cuffquant against the cuffmerge output

cuffquant -o ./cuffquantlib1 -p 12 ./cuffmerge/merged.gtf
lib1TH/accepted_hits.sam
cuffquant -o ./cuffquantlib2 -p 12 ./cuffmerge/merged.gtf
lib2TH/accepted_hits.sam


##Start cuffnorm and cuffdiff

cuffdiff -p 12 -o ./cuffdiff -b
/proj/b2014195/nobackup/data/A26/indexedData/A26.fa --dispersion-method
pooled ./cuffmerge/merged.gtf ./cuffquantlib1/abundances.cxb
./cuffquantlib2/abundances.cxb
```

Script for back-tracing consensus identifiers to gene names

```
#!/usr/bin/perl

## Provide the cuffdiff gene_exp.diff file first, followed by cuffmerged GTF
and finally general GTF annotation file

# Strict and warnings are recommended.

use strict;
use warnings;
use Scalar::Util qw(looks_like_number);

## global vars

my @xlocArray;
my @pvalArray;
my @signArray;
my @funcArray;
my @hypoArray;

## Step 1: Gather significant hits from cuffdiff

my $filename = $ARGV[0];
open(my $fh, $filename) or die "Could not open file '$filename' $!";

while (my $row = <$fh>) {
      chomp $row;
      my @splitrow = split('\t', $row);

#Retrieves hits with a p-value under 0.05

      if(!looks_like_number($splitrow(17)) || $splitrow(17) <= 0.05 ) {
            push (@xlocArray, $splitrow[0]);
            push (@pvalArray, $splitrow(17));
            push (@signArray, $splitrow(39));
      }
}
close $fh or die $1;


##Step 2: Add peg to significant xlocs

$filename = $ARGV(4);
open(my $gtf, $filename) or die "Could not open file '$filename' $!";my
@pegArray = @xlocArray;
$pegArray[0]= "Peg";

while (my $row = <$gtf>) {
      chomp $row;
      my @splitrow = split('\"', $row);
```

```
	for (my $i = 1; $i < @xlocArray; $i++) {
		if($xlocArray[$i] eq $splitrow(4)) {
			$pegArray[$i] = $splitrow(11);
		}
	}
}
close $gtf or die;


##Step 3: Translate peg to function

$filename = $ARGV(9);
open(my $anno, $filename) or die "Could not open file '$filename' $!";
@funcArray = @pegArray;
$funcArray[0]= "Function";


while (my $raw = <$anno>) {
	chomp $raw;
	if($raw =~ "\=") {
		my @splitraw = split('\=', $raw);
		my @splitrew = split('\(', $splitraw(9));
		my $function = $splitrew[0];

		#Extract peg name from longer line

		my @splitPeg = split('\;', $splitraw(4));
		my $pegname = $splitPeg[0];

		for (my $i = 1; $i < @pegArray; $i++) {
			#print "$pegname\n";
			  if(defined($pegname) && $pegname eq $pegArray[$i]) {
				$funcArray[$i]= $function;
			  }
		}
	}
}
close $anno or die;


##Step 3*: Seperate hypotheticals

push(@hypoArray, "Hypotheteticals");
for (my $i = 1; $i < @funcArray; $i++) {
	if($funcArray[$i] =~/hypothetical/) {
		push(@hypoArray, $funcArray[$i]);
		$funcArray[$i] = undef;
```

```
      }
}


##Step 4: Create outfiles

open(OUT, ">cuffDataRetrieve.out") or die;
for (my $i = 0; $i < @xlocArray; $i++) {
      if(defined($funcArray[$i])) {
            print OUT
"$xlocAr-
ray[$i]\t$pvalArray[$i]\t$signArray[$i]\t$pegArray[$i]\t$funcArray[$i]\n";
      }
}

open(OUTHYPO, ">cuffDataRetrieveHypo.out") or die;
print OUTHYPO
"$xlocArray[0]\t$pvalArray[0]\t$signArray[0]\t$pegArray[0]\t$hypoArray[0]\n";

for (my $i = 0; $i < @hypoArray; $i++) {
      if(!defined($funcArray[$i])) {
            print OUTHYPO
"$xlocAr-
ray[$i]\t$pvalArray[$i]\t$signArray[$i]\t$pegArray[$i]\t$hypoArray[$i]\n";
      }
}
```

Production of Circos ready files from Cufflinks output

```
#!/usr/bin/perl

## Provide the cuffdiff gene_exp.diff file
# Strict and warnings are recommended.

use strict;
use warnings;
use Scalar::Util qw(looks_like_number);

## global vars
my @xlocArray;
my @signArray;
my @lposArr;
my @rposArr;


## Step 1: Gather XLOCs and POS

my $filename = $ARGV[0];
open(my $fh, $filename) or die "Could not open file '$filename' $!";

while (my $row = <$fh>) {
      chomp $row;
      my @splitrow = split('\t', $row);

#Retrieve hits with a p-value under 0.05

      if(looks_like_number($splitrow(17))){
            if($splitrow(17) <= 0.05 ) {
                  push (@xlocArray, $splitrow[0]);
                  push (@signArray, $splitrow(39));

                  my @posSplit = split('\:', $splitrow(10));
                  my @positions = split('\-',$posSplit(4));
                  push(@lposArr, $positions[0]);
                  push(@rposArr, $positions(4));
            }
      }
}
close $fh or die $1;

open(OUT, ">A26karyotype.out.txt") or die;
print OUT "chr\t-\ths1\t1\t0\t5790027\tlgrey\n";

for (my $i = 0; $i < @xlocArray; $i++) {
      if($signArray[$i] eq "yes"){
            print OUT
"band\ths1\t$xlocArray[$i]\t$xlocArray[$i]\t$lposArr[$i]\t$rposArr[$i]\tgneg\
n";
```

```
      } if($signArray[$i] eq "no"){
            print OUT
"band\ths1\t$xlocArray[$i]\t$xlocArray[$i]\t$lposArr[$i]\t$rposArr[$i]\tgpos\
n";
      }
}
close OUT or die $1;
```

Generation of the Circos graph

Circos.conf

```
<<include etc/colors_fonts_patterns.conf>>
<<include ideogram.conf>>
<<include ticks.conf>>

<image>
<<include etc/image.conf>>
</image>

chromosomes_units           = 300000
chromosomes_display_default = yes
karyotype = data/karyotype/A26karyotype.out.txt
<<include etc/housekeeping.conf>>
```

Bands.conf

```
show_bands            = yes
fill_bands            = yes
band_stroke_thickness = 15p
band_stroke_color     = lred
band_transparency     = 10
```

Ideogram.conf

```
<ideogram>
<spacing>
default = 0.0025r
break   = 0.5r
</spacing>

<<include ideogram.position.conf>>
<<include bands.conf>>
</ideogram>
```

Ideogram.position.conf

```
radius           = 0.90r
thickness        = 800p
fill             = yes
fill_color       = black
stroke_thickness = 2
stroke_color     = black
```

Ticks.conf

```
show_ticks          = yes
show_tick_labels    = yes

<ticks>
skip_first_label = no
skip_last_label  = no
radius           = dims(ideogram,radius_outer)
tick_separation  = 2p
label_separation = 5p
multiplier       = 1e-5
color            = black
thickness        = 4p
size             = 20p

<tick>
spacing        = 1u
show_label     = yes
label_size      = 30p
label_offset    = 5p
thickness      = 4p
color          = black
size    = 20p
</tick>

<tick>
spacing        = 0.1u
show_label     = no
label_size      = 30p
thickness      = 2p
color          = black
size    = 12p
</tick>

<tick>
spacing        = 0.02u
show_label     = no
label_size      =10p
thickness      = 1p
color          = black
size    = 8p
</tick>

<tick>
spacing        = 5u
show_label     = yes
label_size     = 60p
```

```
label_offset   = 12p
format         = %d
grid           = yes
grid_color     = dgrey
grid_thickness = 1p
grid_start     = 0.5r
grid_end       = 0.999r
size =  32p
</tick>

</ticks>
```

Simplified MAKER tutorial

Introduction

MAKER is an (almost) fully automated application for genome annotation. It easily integrates EST and protein homology data from public repositories to be used with a dozen of different prediction programs with different applications.

In this tutorial we will do two major exercises:

1) Dry run of MAKER with the prepackaged data
2) A full run of MAKER with fresh EST and protein data downloaded from NCBI

   If you have time, you may also try to quickly visualize the output.

The MAKER dry run

Log onto hpc.ilri.cgiar.org with mobaXterm

```
module load maker/2.28
cd ~
mkdir makerTutDry
cd makerTutDry
maker –CTL
ls
```

The maker –CTL runs the MAKER software to produce template versions of the configuration files. These must be changed so MAKER knows what type of analysis you want to use.

`nano maker_opts.ctl and change into the following lines:`

```
genome=/export/apps/maker/2.28/data/dpp_contig.fasta
est=/export/apps/maker/2.28/data/dpp_est.fasta
protein=/export/apps/maker/2.28/data/dpp_protein.fasta
```

Save and exit.

`nano maker_bopts.ctl`

Make sure the settings look reasonable for the software in this file. If you want you may change the numbers here to get affect the quality of your results.

Save and exit.

Run:

`maker maker_bopts.ctl maker_exe.ctl maker_opts.ctl`

Once MAKER finishes running, first check the log file:

`less ~/makerTutDry/dpp_contig.maker.output/dpp_contig_master_datastore_index.log`

If everything worked as intended you should see:

```
contig-dpp-500-500      dpp_contig_datastore/05/1F/contig-dpp-500-500/  STARTED
contig-dpp-500-500      dpp_contig_datastore/05/1F/contig-dpp-500-500/  FINISHED
```

The final output of MAKER has been saved as a .gff file; view it with:
PLEASE NOTE THAT <TAB><TAB><TAB> MEANS YOU SHOULD PRESS TAB THRICE AT THAT POINT, NOT WRITE IT.

```
cd ~/makerTutDry/dpp_contig.maker.output/
less dpp_contig_datastore/<TAB><TAB><TAB>contig-dpp-500-500.gff
```

As you can see maker produces all output very nicely and intuitively; which can easily be exported into a genome browser for visualization.

Full run

In order to simulate a real run of MAKER we will download a small genome, EST and proteins for the species as well as EST from related organisms. For your convenience this data has been prepackaged and can be gotten with:

```
cd ~
wget http://hpc.ilri.cgiar.org/~isylvin/mycgen.zip
unzip mycgen.zip
ls
```

The files in question come from NCBI from the following sites (with some searching of course). It would be very easy for you to fetch the data yourselves in a normal case; but since we are running on HPC it's a bit trickier. We still provide the links so you see how easy it is to fetch supporting data. In case you're curious the data concerns the organism Mycoplasma Genitalium.

http://www.ncbi.nlm.nih.gov/nuccore/108885074?report=fasta Genome

http://www.ncbi.nlm.nih.gov/nucest Related EST

http://www.ncbi.nlm.nih.gov/nucest/?term=Mycoplasma%20genitalium Mycoplasma EST

http://www.ncbi.nlm.nih.gov/protein Protein

Since we are doing a new run we need new option files.

```
module load maker/2.28
cd ~
mkdir makerTutFull
cd makerTutFull
maker -CTL
ls
```

`nano maker_opts.ctl` and change the following lines where **XX** is replaced with the number next to your username. So if you're user20 the first line would be genome=/home/**user20**/mycgen/genMycGen.fasta:

```
genome=/home/user**XX**/mycgen/genMycGen.fasta
organism_type=prokaryotic
est=/home/user**XX**/mycgen/estMycGen.fasta
altest=/home/user**XX**/mycgen/altEstMycGen.fasta
```

```
protein=/home/userXX/mycgen/protMycGen.fasta
model_org=
prok_rm=1
cpus=3
```

Save and exit. Run:

```
interactive
maker maker_bopts.ctl maker_exe.ctl maker_opts.ctl
```

Analysis should take between five and ten minutes. Once MAKER finishes running, first check the log file:

```
less ~/makerTutFull/genMycGen.maker.output/genMycGen_master_datastore_index.log
```

If everything worked as intended you should see:

```
contig-dpp-500-500      dpp_contig_datastore/05/1F/contig-dpp-500-500/  STARTED
contig-dpp-500-500      dpp_contig_datastore/05/1F/contig-dpp-500-500/  FINISHED
```

The final output of MAKER has been saved as a .gff file; view it with:

```
cd ~/makerTutFull/genMycGen.maker.output/
less genMycGen_datastore/<TAB><TAB<TAB> gi%7C108885074%7Cref%7CNC_000908%2E2%7C.gff
```

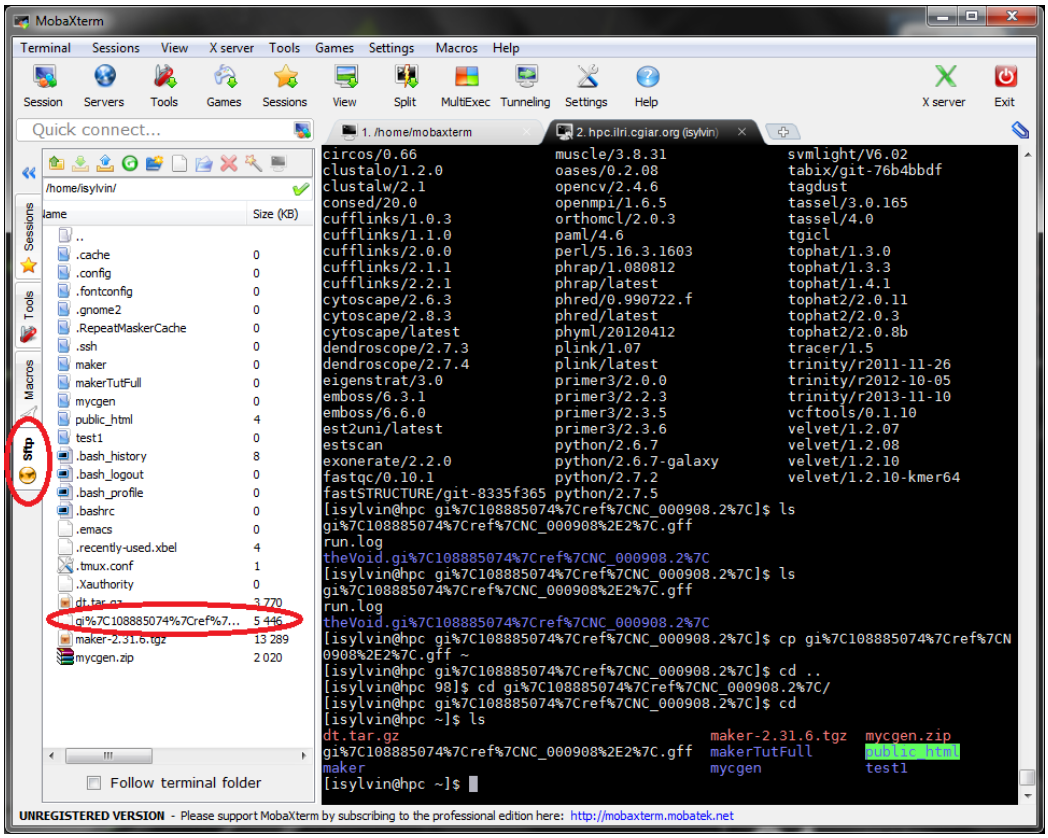
If You Have Time: Visualize your data

If you have time you might want to look at your now annotated data through a genome browser; the following can be done very quickly if you're interested.

Do the following on one line (not two like in this document).

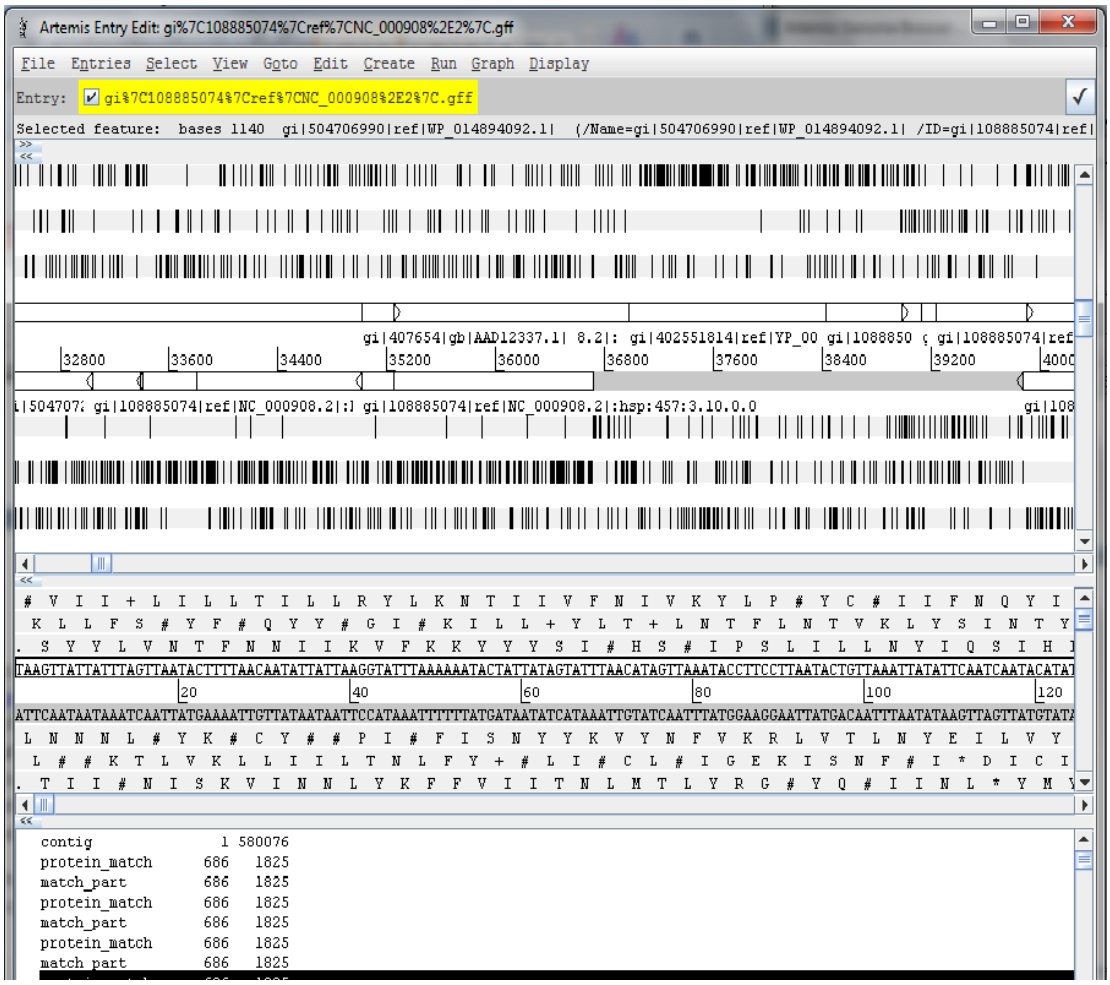
```
cp ~/makerTutFull/genMycGen.maker.output/genMycGen_datastore/<TAB><TAB<TAB>
gi%7C108885074%7Cref%7CNC_000908%2E2%7C.gff ~
```

You then have to copy the file to your desktop. Click on the SFTP tab in moba terminal; rightclick in the white box with all the files and select "Refresh current directory". Then drag the file named gi%7C108885074%7Cref%7CNC_000908%2E2%7C.gff from your white window and onto your desktop. **An image is attached to help you understand the process:**



Once on your desktop, download Artemis from https://www.sanger.ac.uk/resources/software/artemis/#download and start it (with a double click on windows, on Linux it needs to be started with the command line `java -jar artemis.jar` whilst in the download folder). Press OK; press File -> Open File Manager and select your file. It will most likely be located under **YOURUSERNAME**`/Desktop/gi%7C108885074%7Cref%7CNC_000908%2E2%7C.gff`
If you get a warning just ignore it and press no.

If you've done everything correctly the genome browser will now produce the unannotated genome you downloaded with the genomes predicted by MAKER (see image).


Artemis Entry Edit: gi%7C108885074%7Cref%7CNC_000908%2E2%7C.gff
File Entries Select View Goto Edit Create Run Graph Display
Entry: gi%7C108885074%7Cref%7CNC_000908%2E2%7C.gff
Selected feature: bases 1140 gi|504706990|ref|WP_014894092.1| (/Name=gi|504706990|ref|WP_014894092.1| /ID=gi|108885074|ref|
32800
33600
34400
35200
36000
36800
37600
38400
39200
4000
TAAGTTATTATTTAGTTAATACTTTTAACAATATTATTAAGGTATTTAAAAAATACTATTATAGTATTTAACATAGTTAAATACCTTCCTTAATACTGTTAAATTATATTCAATCAATACATAT
20
40
60
80
100
120
ATTCAATAATAAATCAATTATGAAAATTGTTATAATAATTCCATAAATTTTTTATGATAATATCATAAATTGTATCAATTTATGGAAGGAATTATGACAATTTAATATAAGTTAGTTATGTATA
contig 1 580076
protein_match 686 1825
match_part 686 1825
protein_match 686 1825
match_part 686 1825
protein_match 686 1825
match_part 686 1825

Simple tutorial for submitting data to NCBI

To what database should I submit my data?

The lecture today will briefly mention where most data types should go. In case you have a very special type of data (or just poor memory) check out the following links:

1. *Navigate to https://submit.ncbi.nlm.nih.gov/ and see if you can figure it out*
2. *Else check out http://www.ncbi.nlm.nih.gov/guide/howto/submit-data/ for more info*

What if I really, really can't find an answer after this course?

Make sure you've really checked the NCBI webpage for help. Maybe you even googled "site: http://www.ncbi.nlm.nih.gov WHAT IS COVERAGE". As a final resort you can mail NCBI at info@ncbi.nlm.nih.gov .

Creating an NCBI account

1. Navigate to NCBI homepage: www.ncbi.nlm.nih.gov
2. Use the "*Sign in to NCBI"* in the top right
3. Register using "*Register for an NCBI account"* hyperlink to the left
4. Open up your e-mail and validate your registration; remember to check your spam folder!

GenBank submission

Submission through BankIt

Start at the submission portal: https://submit.ncbi.nlm.nih.gov/ . Click the Genbank link followed by the BankIt link. Then press the "Sign in to use BankIt" link in the top right.

Note: Sequence must be in fasta (not fastq) and include definition lines.

>Seq2 [organism=Mus musculus] ***Mouse strain BMC2/3 cytochrome b (cytb) complete CDS***
ttatatcgatatgacacccgggatatacagatattagggata

Definition line is featured in bold. Failure provide them will mean the submission will either be delayed or outright refused.

Use *"Sign in to use BankIt"*, top-left

If you have an account, log in with it, otherwise see "*How do I get a NCBI account?"*

Click *"New submissions"*. The following provide some outlines for each tab:

Contact

Don't need to fill in all fields, like fax numbers. Notice that BankIt asks you to press continue again even if the fields you skip are optional.

Reference

Add authors and at least the title of the paper to write.

Sequencing tech

Assembly name is optional, NCBI adds its own if you don't provide one. You should be able to find the coverage of your sequence by either looking at the output of the assembler, or tallying with a bash script.

Nucleotide

Molecule type, topology and genomic completeness are all very important but luckily very intuitive. Don't worry about providing a correct nucleotide sequence amount as BankIt counts it for you in case you leave it blank.

Submission category

Either you created the data from scratch or you improved someone's work.

Source modifiers

Organelle/Location is per default genomic and doesn't need to be selected.

Add as many as source modifiers as possible. Clicking the optional 'include primers' checkbox opens up the primers taö. You can use a tab delimited file (if data varies between samples, or you don't want to use the web form). This is an example tab delimited file:

Sequence_ID	Specimen_voucher	Collected_by	Collection_date	Country	Identified_by	Lat_Lon
Seq1	MKP 334	C. Grant	31-Jan-2001	USA	C. Grant	13.57 N 24.68 W
Seq2	MKP 1230	S. Tracy	28-Feb-2002	Slovakia	C. Grant	13.24 N 24.35 W

Commonly used Source Modifiers

- **Clone** - Name of clone from which sequence was obtained.
- **Collection_date** - Date the specimen was collected.
  In format **DD-Mon-YYYY**, that is 2-digit date, three-character abbreviation of month, and 4-digit year, (*e.g.*, 11-Feb-2002).
  Mon-YYYY and **YYYY** are alternate formats to use when date information is less complete.
- **Country** - The country where the sequence's organism was located. May also be an ocean or major sea. Additional region or locality information must be after the country name and separated by a ':'. For example: USA: Riverview Park, Ripkentown, MD
- **Host** - When the sequence submission is from an organism that exists in a symbiotic, parasitic, or other special relationship with some second organism, the 'host' modifier can be used to identify the name of the host species.
- **Isolate** - Identification or description of the specific individual from which this sequence was obtained.
- **Isolation source** - Describes the local geographical source of the organism from which the sequence was obtained.

- **Specimen_voucher** - An identifier of the individual or collection of the source organism and the place where it is currently stored, usually an institution.

This should be provided using the following format 'institution-code:collection-code:specimen-id'. specimen-id is mandatory, collection-code is optional; institution-code is mandatory when collection-code is provided. Examples:

    - 99-SRNP
    - UAM:Mamm:52179
    - personal collection:Joe Smith:99-SRNP
    - AMCC:101706

- **Strain** - Strain of organism from which sequence was obtained.

The following source modifiers are available to further describe the sequences in a BankIt set:

- **Altitude** - Altitude in metres above or below sea level of where the sample was collected.
- **Authority** - The author or authors of the organism name from which sequence was obtained.
- **Bio_material** - An identifier for the biological material from which the nucleotide sequence was obtained, with optional institution code and collection code for the place where it is currently stored.

This should be provided using the following format **'institution-code:collection-code:material_id'**. material_id is mandatory, institution-code and collection-code are optional; institution-code is mandatory when collection-code is present.

This qualifier should be used to annotate the identifiers of material in biological collections which include zoos and aquaria, stock centers, seed banks, germplasm repositories and DNA banks.

- **Biotype** - Variety of a species (usually a fungus, bacteria, or virus) characterized by some specific biological property (often geographical, ecological, or physiological). Same as biotype.
- **Biovar** - See biotype
- **Breed** - The named breed from which sequence was obtained (usually applied to domesticated mammals).
- **Cell_line** - Cell line from which sequence was obtained.
- **Cell_type** - Type of cell from which sequence was obtained.
- **Chemovar** - Variety of a species (usually a fungus, bacteria, or virus) characterized by its biochemical properties.
- **Clone** - Name of clone from which sequence was obtained.

- **Collected_by** - Name of person who collected the sample.
- **Collection_date** - Date the specimen was collected.
  In format **DD-Mon-YYYY**, that is 2-digit date, three-character abbreviation of month, and 4-digit year, (*e.g.*, 11-Feb-2002).
  Mon-YYYY and **YYYY** are alternate formats to use when date information is less complete.
- **Country** - The country where the sequence's organism was located. May also be an ocean or major sea. Additional region or locality information must be after the country name and separated by a ':'. For example: USA: Riverview Park, Ripkentown, MD
- **Cultivar** - Cultivated variety of plant from which sequence was obtained.
- **Culture_collection** - Institution code and identifier for the culture from which the nucleotide sequence was obtained, with optional collection code.

This should be provided using the following format **'institution-code:collection-code:culture-id'**. culture-id and institution-code are mandatory.

This qualifier should be used to annotate live microbial and viral cultures, and cell lines that have been deposited in curated culture collections.

- **Dev_stage** - Developmental stage of organism.
- **Ecotype** - The named ecotype (population adapted to a local habitat) from which sequence was obtained (customarily applied to populations of Arabidopsis thaliana).
- **Forma** - The forma (lowest taxonomic unit governed by the nomenclatural codes) of organism from which sequence was obtained. This term is usually applied to plants and fungi.
- **Forma_specialis** - The physiologically distinct form from which sequence was obtained (usually restricted to certain parasitic fungi).
- **Fwd_primer_name** - name of forward PCR primer
- **Fwd_primer_seq** - nucleotide sequence of forward PCR primer
- **Genotype** - Genotype of the organism.
- **Haplogroup** - Name for a group of similar haplotypes that share some sequence variation
- **Haplotype** - Haplotype of the organism.
- **Host** - When the sequence submission is from an organism that exists in a symbiotic, parasitic, or other special relationship with some second organism, the 'host' modifier can be used to identify the name of the host species.

- **Identified_by** - name of the person or persons who identified by taxonomic name the organism from which the sequence was obtained
- **Isolate** - Identification or description of the specific individual from which this sequence was obtained.
- **Isolation source** - Describes the local geographical source of the organism from which the sequence was obtained.
- **Lab_host** - Laboratory host used to propagate the organism from which the sequence was obtained.
- **Lat_Lon** - Latitude and longitude, in decimal degrees, of where the sample was collected.
- **Note** - Any additional information that you wish to provide about the sequence.
- **Pathovar** - Variety of a species (usually a fungus, bacteria or virus) characterized by the biological target of the pathogen. Examples include Pseudomonas syringae pathovar tomato and Pseudomonas syringae pathovar tabaci.
- **Pop_variant** - name of the population variant from which the sequence was obtained
- **Rev_primer_name** - name of reverse PCR primer
- **Rev_primer_seq** - nucleotide sequence of reverse PCR primer
- **Specimen_voucher** - An identifier of the individual or collection of the source organism and the place where it is currently stored, usually an institution.

This should be provided using the following format 'institution-code:collection-code:specimen-id'. specimen-id is mandatory, collection-code is optional; institution-code is mandatory when collection-code is provided. Examples:

    - 99-SRNP
    - UAM:Mamm:52179
    - personal collection:Joe Smith:99-SRNP
    - AMCC:101706

- **Serogroup** - Variety of a species (usually a fungus, bacteria, or virus) characterized by its antigenic properties. Same as serogroup and serovar.
- **Serotype** - See Serogroup
- **Serovar** - See Serogroup
- **Sex** - Sex of the organism from which the sequence was obtained.

- **Strain** - Strain of organism from which sequence was obtained.
- **Sub_species** - Subspecies of organism from which sequence was obtained.
- **Subclone** - Name of subclone from which sequence was obtained.
- **Subtype** - Subtype of organism from which sequence was obtained.
- **Substrain** - Sub-strain of organism from which sequence was obtained.
- **Tissue_lib** - Tissue library from which the sequence was obtained.
- **Tissue_type** - Type of tissue from which sequence was obtained.
- **Type** - Type of organism from which sequence was obtained.
- **Variety** - Variety of organism from which sequence was obtained.

Primers

Different reaction sets = Primers separated by multiple reactions

Features

Features should be provided in a document in Plain ASCII. It can be provided in the web form but it's really, really tedious.

Every sequence is divided into sections. Every row is a pair of identifier and then value.

If a feature is reversed, so are the indexes.

< > means incomplete (partial features) meaning they start and stop upsteams and downstreams of the nucleotide positions respectively.

All genes should include a gene index which is positioned so gene = 5'UTR+CDS+3'UTR.

If you get unsure about how to annotate something you can always mail info@ncbi.nlm.nih.gov .

>Feature Seq1
<1 >1050 gene
gene ATH1
<1 1009 CDS
product acid trehalase
product Athlp
codon_start 2
<1 >1050 mRNA
product acid trehalase

>Feature Seq2
2626 2590 tRNA
2570 2535

product tRNA-Phe

>Feature Seq3
1080 1210 CDS
1275 1315
product actin
note alternatively spliced
1055 1210 mRNA
1275 1340
product actin
1055 1340 gene
gene ACT
1055 1079 5'UTR
1316 1340 3'UTR

Review and correct

You may download your complete set as a zip file.

Do not press "Finish submission" as it sends your test to NCBI!

Finally in case you haven't received an automatic reply, your genbank accession number or final records you can always mail gb-admin@ncbi.nlm.nih.gov to see the status of your project.

Sequin

Sequin can be used locally on your machine. Just download

http://www.ncbi.nlm.nih.gov/Sequin/download/seq_download.html

make a directory and move the file there and doubleclick on the sequin executable. A bunch of files should be generated. Start sequin.exe. If you need a fasta file to use as template you can download one from http://hpc.ilri.cgiar.org/~isylvin/seqFasta.fasta

Metadata creation

BioSample

Start at the submission portal: https://submit.ncbi.nlm.nih.gov/ . Click the BioSample link.
In the future in case you want to create multiple BioSamples at once there's a link to download a "batch template". Open the file in Excel and add info to it, at least to all columns marked with an *.
If you're unsure about what to put in each field, use https://submit.ncbi.nlm.nih.gov/biosample/template/?package=MIGS.eu.human-associated.4.0&action=definition as a reference.

Sample type

Choose "Genome, metagenome or marker sequences" per default. Just make sure your data is MIxS compliant (Minimum information about (x) sequences).

Attributes

Pick something nice for sample name.

For "isolation and growth condition" you'll be needing a PMID or similar URL for the protocol/SOP.

The ”reference for biomaterial” requires an uploaded report as well.

For “geographic location” most country names exist. However, a full list is located here: *http://www.insdc.org/documents/country-qualifier-vocabulary*

Overview

Make sure you don’t hit SUBMIT as we’re mostly fooling around with our entries

BioProject

Start at the submission portal: https://submit.ncbi.nlm.nih.gov/ . Click the BioProject link.

Please make sure to create a BioSample before you start on a BioProject.

Use https://submit.ncbi.nlm.nih.gov/ and select “BioProject”

Submitter

Boring and trivial. Just make sure you get the organization and department right.

Project type/ Target

It is highly suggested to use Google to find the definitions of the terms you’re unsure of. Even if the data is submitted properly, inputting incorrect classifications into the necessary data might poison further studies.

Make sure you fill in as many optional fields as possible during a real run.

BioSample

Use the format SUBxxxxx: BIOSAMPLENAME.

Publications

One of the few steps that is actually validated. Use a PubMed id like “25107883” to continue.

A doi is very similar to an ISBN, and is much more general than a PubMed id (PMID).

Overview

Make sure you don’t hit SUBMIT as we’re mostly fooling around with our entries

Submitting GEO data

Start at the submission portal: https://submit.ncbi.nlm.nih.gov/ . Click the microarray GEO link.

If you ever need professional help you can mail geo@ncbi.nlm.nih.gov .

First of all press the “Submit” button on the page. Then upload the data and fill out the form. The preferred format is GEOarchive.

Make sure you don’t hit SUBMIT as we’re mostly fooling around with our entries

If you want to doublecheck your SOFT or MiniML formatted data, use http://www.ncbi.nlm.nih.gov/geo/submission/depslip.cgi?subm=0 and actually submit to it to test your format.

Submitting SRA data

Start at the submission portal: https://submit.ncbi.nlm.nih.gov/ . Click the SRA link. Bear in mind all SRA studies need a BioProject or at least an associated BioSample.

Use either login route, preferably NIH. Sometimes the login is down. Enter an alias (project description) and possibly an internal comment.

Click “Set new experiment”

Alias is the experiment name; title is used to call out individual records from the experiment.

Add library info about how the data was sequenced.

Pipeline refers to **all** the bioinformatical programs used to manipulate the data.

Links and attributes lets you add links like DDBJ.

Save then click new run. Add new files in a format like fastq or bam.

Teaching material used at SLU, Sweden

Summary

This guide will cover the following areas:

1) Installing Ubuntu Linux
2) Traversing Linux through the command line
3) Understanding annotation pipelines
4) Understanding MAKER
5) Installation of the MAKER software suite
6) Running the MAKER software suite

Some of the contents of this may be altered based on previous knowledge and interest in specific fields.

Ubuntu installation

What is Ubuntu and Linux?

Click the following link and digest the contents of it. There will be a lot of more external resources, so make sure you take your time to fully comprehend each step.

http://computer.howstuffworks.com/ubuntu.htm

How do I get Ubuntu?

Any serious bioinformatician will quickly realize that most bioinformatical tools are written by scientists on Linux – for Linux. As such we will familiarize ourselves with installing Linux. **Although we won't be installing any new operating systems as of now**, it is very good to be accustomed to the procedure.

https://www.youtube.com/watch?v=a7041b90QpY

The video (courtesy of Christopher Barnatt) outlines how to install Ubuntu with your own bootable Ubuntu disk. There are many other variations of both Linux and installing Linux. There are for instance the possibility to use an USB stick to install Ubuntu using software downloaded here:

http://unetbootin.sourceforge.net/

We won't be installing Ubuntu right now, mostly because it removes all the old data unless one uses proper formatting. This is however an excellent time to familiarize oneself with the procedure, as it has been almost identical for several years.

Traversing Linux with the command line

With Ubuntu it is possible to rely on the same methods as those used to move about in Windows. However, the second one tries to use bioinformatics one will undoubtedly run into some problems. More so if one working against a server station, which only allows command line input, and has never used the command line in the past.

As such it is important to get familiarizes with the command line and the power behind it. For this, William E. Shotts, Jr. has assembled a great tutorial freely available online:

http://linuxcommand.org/lc3_learning_the_shell.php#contents

In the case a Linux PC isn't available we will use CygWin to emulate Linux commands on the Windows machine. Some part of the tutorial may not function perfectly, but the results should be close enough.

As such, if you're using a Windows machine, download and install CygWin from:
https://cygwin.com/install.html

If the download is slow, continue with the document and return to the command line tutorial once it finishes.

Understanding annotation pipelines

An article was published in Nature two years ago that outlines the thinking and general ideas for an annotation project. This is one of the few articles that focuses on the thinking rather than the software solutions. With that said all the solutions are still current as of this date. It is very well worth the read:

http://www.nature.com/nrg/journal/v13/n5/full/nrg3174.html

Understanding Maker

We have previously gone through basic functionality of GMOD's MAKER software suite. Now we will focus on understanding how MAKER works on a more theoretical level. Read sections 2, 3 and 6 of the following document; the other sections are still a good read but might be a bit too specific to be applicable as of yet.

http://gmod.org/wiki/MAKER_Tutorial

Installation of MAKER

Installation of MAKER, as we have yet to secure a Linux machine, will be done on the planetsmasher (or UPPMAX) cluster as an interactive demo/tutorial. The installation is expected to not just be smooth sailing and will serve to highlight some problems with server installations and how to resolve them.

Running the MAKER software suite

The MAKER hands-on has previously been done in the workshop in Nairobi. We will use this as a basis for another demo/tutorial run; but this time focus on all the details essential for running MAKER properly.