Comparative analysis of Gene Finding tools when applied to *Trichoderma* genomes

A thesis submitted to the
College of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements
for the degree of Master of Science
in the Applied Computing of Computer Science
University of Saskatchewan
Saskatoon

By
Connor Burbridge, Dave Schneider, Tony Kusalik

Permission to Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Disclaimer

Reference in this thesis to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan, and shall not be used for advertising or product endorsement purposes.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building, 110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

Abstract

placeholder

Acknowledgements

I would like to acknoweldge both Dr. Kusalik and Dave Schneider for their excellent support and mentorship during both COVID and my MSc. project. I would also like to thank Brendan Ashby and Dr. Leon Kochian for providing both data and additional financial support.

Contents

Permission to Use . . . . . **i**

Abstract . . . . . **ii**

Acknowledgements . . . . . **iii**

Contents . . . . . **iv**

List of Tables . . . . . **vi**

List of Figures . . . . . **vii**

List of Abbreviations . . . . . **viii**

1 Introduction **1**

2 Background **2**
2.1 Genomics 2
2.1.1 Sequencing 2
2.2 Whole Genome Shotgun Sequencing 2
2.3 Next Generation Sequencing - Illumina 3
2.4 3rd Generation Sequencing - Nanopore 3
2.5 Trichoderma 3
2.6 Genome Assembly 4
2.7 Identification of AT-rich Genomic regions 5
2.8 Repeat Identification and Masking 5
2.9 Centromere Identification 6
2.10 Gene Finding Methods 6
2.11 RefSeq 7
2.12 InterProScan 8
2.13 File Formats 8
2.13.1 FASTA 8
2.13.2 FASTQ 8
2.13.3 General Feature Format - GFF 9

3 Research Questions **11**
3.1 Research Questions 11
3.2 *Trichoderma* Assembly Results 11
3.3 Profiling of Gene Finding Tools 11
3.4 Number of Features Predicted 12
3.5 Lengths of Predicted Genes 12
3.6 Performance in Regions of Anomalous Sequence Content 12
3.7 BUSCO Completeness 12
3.8 Identifying Regions of Agreement and Disagreement 12
3.9 Validation of Predicted Genes via InterProScan 13
3.10 Identification of a Core *Trichoderma* Gene Set. 13
3.11 Selection of a Gene Finding Tool 13

4 Data and Methodology **14**
4.1 Workflow Overview 14

4.2 Sequence Pre-processing and Assembly . . . 14
4.3 Gene Prediction Using GeneMark-ES . . . 14
4.4 Gene Prediction Using Braker2 . . . 14
4.5 Binomial Tests on CDS Lengths . . . 16
4.6 Evaluating Gene-Finder Performance Using BUSCO . . . 16
4.7 Region Identification Procedure . . . 16
4.8 Functional Annotation Using InterProScan . . . 16
4.9 Identification of AT-rich Genomic Windows . . . 18
4.10 Ground-truthing with BLAST . . . 18
4.11 Visualization of Identified Regions . . . 18

5 Results . . . 19
5.1 Assemblies of DC1 and Tsth20 . . . 19
5.2 Installation and Profiling Gene Finding Tools . . . 21
5.3 Initial Gene Finding Results . . . 22
5.4 Distribution of Predicted Coding Sequence Lengths . . . 23
5.5 BLAST Results . . . 27
5.6 BUSCO Results . . . 29
5.7 Region Identification . . . 31
5.8 Genes in Regions of Anomalous GC Content . . . 35
5.9 InterProScan as Supporting Evidence for Predicted Genes . . . 37
5.10 Selection of Gene Finding Tool . . . 40
5.11 Conclusions . . . 43

References . . . 44

List of Tables

5.1 General assembly metrics produced by QUAST[15] (a genome quality assement tool). 19
5.2 Gene prediction counts 23
5.3 Table of P-values from two-sided two-sample Kolmogorov-Smirnov tests between gene finding tools. 26
5.4 tblastn hits from reference protein sequences to selected *Trichoderma* assemblies. Hits are reported if the alignment length is greater than 30% of the reference protein length and if 30% of the aligment length has identical matches. 27
5.5 Counts of transcripts predicted by Braker2, GeneMark and RefSeq in regions containing tblastn hits from *Ttrichoderma atroviride*, *Fusarium granminarium*, and *Saccharomyces cerevisiae*. . 28
5.6 Results from BUSCO using the fungal analysis option organized by gene finding tool. The selected BUSCO dataste contains 758 markers. For more information on the categories assigned by BUSCO, please refer to the documentation. 29
5.7 GeneMark missed BUSCO proteins 30
5.8 Agreement of predictions in anomalous GC regions. 36
5.9 Number of genes predicted by Braker2, GeneMark and RefSeq in AT-rich genomic sequence from each assembly. It is important to note that in the cases of DC1 and Tsth20, full support indicates supporting gene predictions from both GeneMark and Braker2. 37
5.10 p-values produced from a two-sided binomial test for each combination of tool and assembly. 37
5.11 InterProScan Pfam Evidence 38
5.12 Final scoring table 42

List of Figures

2.1 Example of two FASTA sequence entries. One example with sequence characters split across multiple lines, and one showing all sequence characters on the same line. 8
2.2 Example of the four lines in a FASTQ entry. 9
2.3 An example of GFF entries for a sinlge gene. 9

4.1 A flowchart of the methodology followed in this research. The workflow is broken up into sections based on the processes and data involved at each step. Oval-shaped nodes represent steps involved in the processing of data, while rectangular nodes represent intermediate datasets and results. 15

5.1 Plots showing the frequency of GC values calculated from sliding windows for each assembly. 20
5.2 CDF plots part 1. 24
5.3 CDF plots part 2. 25
5.4 CDF plots part 3. 26
5.5 Breakdown of identified regions 33
5.6 Example of a region with many gene calls 34
5.7 Predictions on opposing strands 35
5.8 Agreeing Pfam matches 38
5.9 Split Pfam matches 39
5.10 RefSeq absence with IPS evidence 40

List of Abbreviations

DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
CDS	Coding sequence
Mb	Megabases
Kb	Kilobase
WGS	Whole Genome Shotgun (sequencing)
NGS	Next generation sequencing
PCR	Polymerase chain reaction
GMO	Genetically modified organism
CPU	Central processing unit
GC	Guanine cytosine
HMM	Hidden Markov model
UTR	Untranslated region
GFF	General feature format
BUSCO	Benchmarking Universal Single-Copy Orthologs
RSMI	Root, Soil and Microbial Interactions
MITE	Minature inverted-repeat transposable element
TIR	Terminal inverted repeat
TDR	Terminal direct repeat
maybe	Should I include tools like BLAST and resources like NCBI?

1 Introduction

The study of organisms in Biology is a highly complex complex process involving many disciplines. To better understand how these organisms function, we must braker down the problem into different sub-problems. One important sub-problem in biology is the understanding of the molecular tools and processes used by cells to function in normal and abnormal environmental scenarios or stress conditions. These conditions may include disease and environmental stress for example. However, previous work in biology has shown that the underlying backbone of information, known as the genome, can vary widely between different organisms in many aspects, such as overall structure, length, ploidy, methylation, and overall gene content. All of these aspects can affect the survival of an organism in any given environment, some of which may be interesting to researchers. As an example, one of the most popular and extensively studied diseases is cancer. Through the study of human genomes, researchers identified a key gene, named TP53, involved in the suppression of tumours. Functional mutations affecting this gene can result in increased risk of cancer. Understanding how and why TP53 suppresses tumours can provide insight into future cancer prevention methods and treatments. These genes can then be mapped to the genome of the organism to identify its location.

This general workflow can be applied to features of interest from organisms in all branches of life. To facilitate this process of identifying genes from a genome, the process of genome annotation was developed. In this case, instead of identifying a gene of interest and then mapping it back to the genome, gene finders identify potential genes from a reference genome before truly knowing their function or if the candidate is truly untilized by the organisms. This set of potential genes acts as a reference for future research. However, to generate a reliable set of possibe genes, a gene finder must be supplied with a suitable high quality genome. In many cases, researchers may be studying a specific strain or variety of organism that differs from the reference assembly and annotation available for the organism of interest. Rather than using the reference assembly for analysis, it may be beneficial to generate a new assembly for the unique variety or strain, which must then be passed through a gene finding tool. With the variety of gene finding tools and approaches, the choice of an appropriate tool can affect the resulting gene set. This problem raises a question. Do the results from gene finding tools differ? And more importantly, how should one compare results from gene finding tools when there is no reference annotation for a specifc variety in question? Most genome annotation tools benchmark their performance in comparison to an existing reference annotation, which is usually considered to be of suitably high quality. This is not possible in the case of unique assemblies, and so the devleopment of a comparative methodology is in order.

2 Background

2.1 Genomics

Genomics is a wide area of study focusing on the genomes of organisms from all varieties of life. A genome is a sequence of characters that contains the fundamental set of 'rules' used to create what we know as life. One can think of a genome as a set of instructions that our cells use in order to complete the tasks that make us function. A genome is comprised of tightly bundled sequences of DNA, which are stored in the nucleus of cells. These bundles of DNA contain sections known as genes, which can be thought of as the tools described by the set of instructions. These tools carry out a vast number of processes ranging from no known function at all to genes that are key in protecting against diseases cancer. Genomes can vary widely in size, ranging from small bacterial genomes of roughly 4 Mb up to approximately 149000 Mb. Piecing together genomes provides numerous opportunities to understand other 'omics' within cells, such as proteomics, metabolomics, transcriptomics and epigenomics.

2.1.1 Sequencing

Sequencing data is a pivotal form of data used in nearly all applications of Bioinformatics. To understand the processes used by organisms for day to day survival or in unique circumstances, we must have an initial set of data points to work with. These sequences, referred to as reads after sequencing, are the foundation for solving problems ranging from taxonomical classification to the understanding or complex biological functions like signaling pathways. Reads may come in a variety of forms and formats depending on the desired application.

2.2 Whole Genome Shotgun Sequencing

Whole Genome Shotgun sequencing (WGS), is a method to produce a large number of genomic sequences from a sample of interest. This is form of sequencing is quite common as it is has a wide variety of applications in research [2]. WGS involves slicing up genomic DNA into smaller segments. These small segments are then processed further resulting in a set of physical molecules that can be supplied to a compatible sequencing platform of which there are a variety. Modern sequencing platforms are comprised of next generation sequencing (NGS) and 3rd generation sequencing approaches.

2.3 Next Generation Sequencing - Illumina

Illumina sequencing is one of the most popular NGS platforms currently available. Illumina sequencing produces a very large number of high quality short reads, typically between 75 and 250 base pairs in length. Sequencing libraries can be prepared to produce reads solely from one end of a sequence fragment (single-end) or both ends (paired-end). Advantages of paired end sequences are the additional context provided by the paired sequence on the opposite end of the fragment. This context is leveraged by read processing tools to identify features such as repetitive regions and genomic rearrangments, which can be significant in downstream analyses. Illumina sequence librairies are generated by first fragmenting the DNA samples, amplifying them via PCR, ligating adapters that allow the sequence to bind to the sequencing plate, and finally identifying each fragment's sequence of nucleotides using fluorescently-labeled nucleotides that bind to the fragments [14].

2.4 3rd Generation Sequencing - Nanopore

Nanopore sequencing data is relatively recent approach to sequencing projects. While Illumina reads are considered to be short, Nanopore reads are much larger, ranging from 10Kb to 300Kb depending on the approach used. Long reads are beneficial due to their ability to bridge the gaps between difficult to assemble regions when performing sequence assembly. An example of a difficult to assemble region would be a region with a high repeat content, where a large number of small repeats may be collapsed during the assembly process, resulting in an assembly that does not represent the true nature of the sequence being studied [21]. Whlie Nanopore was previously known for having lower quality base calls when compared to Illumina, that is no longer the case at this time. Nanopore sequencing works by passing long seqments of genetic sequence through a membrane bound protein and measuring changes in electrical current, which is characteristic of the nucleotide at a given position.

2.5 Trichoderma

Crop resistance to environmental stressors is a necessity for crop health and overall crop yields. Current popular methods for crop protection involve the use of pesticides and genetically modified organisms, which can be expensive and potentially politically dividing in the case of GMOs[35]. In addition, crops suffer when soils are not sufficient for crop growth and health. Soil insufficiencies can result in drought stress as well as nutrient stress, leading to poor overall yields.

Trichoderma is a fungi that can both communicate with and colonize the roots of plants in a non-toxic, non-lethal, opportunistic symbiotic relationship[39]. Many strains of *Trichoderma* have been shown to provide resistance to pathogenic bacteria and other fungi in soils through the use of polyketides, non-ribosomal peptide

synthetases and other antibiotic products[39]. Recently, two strains of *Trichoderma* have been identified in the prairie regions of Alberta and Saskatchewan. These two strains, named Tsth20 and DC1, have been found to have beneficial properties when used as an inoculant for plants in the soils mentioned before. In addition to these beneficial properties, the two strains mentioned previously provide even further protection for plants in dry, salty soils and one strain also has potential for use as a bioremediation tool in soils contaminated with hydrocarbon content. Bioremediation and resistance to drought tolerance has also been investigated in other strains of *Trichoderma* as well[31]. However, little is known about the mechanisms at work in these strains, so DC1 and Tsth20 were sequenced by the Global Institute for Food Security (no publication yet) in an initial attempt to better understand the details of these genomes. While this research does not directly identify genomic elements related to the secretome of these genomes, it may serve as a foundation for future research of *Trichoderma*.

2.6 Genome Assembly

Sequence assembly has been a long-standing problem in the field of bioinformatics[22]. Determining the correct order and combination of smaller subsequences into an accurate complete sequence assembly is computationally difficult in terms of compute resources such as memory, CPU cycles and storage required for input sequences[22]. In addition to these difficulties, there can be other issues encountered during asssembly due to the nature of the data or genomes themselves, such as low quality base calls for long read data, which is not necessarily the case today, or the inherent content of genomes themselves using repetitive regions as an example. Insufficient data used in an assembly may result in short, fragmented assemblies, depending on the size of the genomes, while sequence data that is not long enough can fail to fully capture repetitive regions in an assembly. To solve this problem, a wide range of assembly tools have been developed with their own unique approaches to the genome assembly problem, so it is important to use an appropriate assembler for the task at hand, and also important to evaluate the assembly thoroughly.

Genome assembly tools generally approach the assembly problem using a graph-based approach. The most common graph-based approach is the de Bruijn graph assembly [10]. A graph in this context, is set of nodes (k-mers from sequences) connected by edges (overlaps between k-mers). Traversing through this graph results in longer subsequences that ultimately result in a set of consensus sequences and final assembly. In the early years of long read sequence data, sequencing platforms encountered difficulties producing consistently high scores for base calls when seuqncing. To combat this, some assembly workflows may also include a polishing or correction step once the initial assembly is completed in which high quality short read sequences are supplied as supplemental information to correct low quality base calls in the assembly. These low quality base calls are typically not present in modern long read sequencing approaches as the methodology and quality of calls have improved drastically. While the polishing step is arguably unnecessary in modern assemblies, the polishing programs remain available should researchers be interested in applying additional reads for

polishing.

One approach to aid in the previously mentioned issue of assembly correctness is to use a combination of long and short reads in what is known as a hybrid assembly. Combining both highly accurate short reads with deep coverage along with less accurate but much longer reads can produce high quality genome assemblies that capture long repetitive regions. Hybrid assembly approaches have been shown to produce high quality assemblies in a wide variety of organisms as the combine long read data with short data to produce assemblies that properly represent long repetetive regions with additionaly high quality Illumina sequences for correction. Once assembled, the sequences must also be evaluated with measures such as N50, L50, coverage, average contig length and total assembled length to ensure that the genomes are well assembled, at least based on these metrics[22]. Following appropriate assembly protocols is essential to the further success of a project as downstream processing such as annotation depends on a high-quality assembly.

2.7 Identification of AT-rich Genomic regions

One important aspect of interest when assembling any form of sequence is GC content or percent GC of the assembled sequence. Large regions of anomolous GC content may be of interest to researchers as they may contain repetitive regions and unique features responsible for traits specific to the organism in question.

2.8 Repeat Identification and Masking

Repeat identification within assembled genomes is a problem that needs to be considered during the genome annotation process. Regions with long repeats can have a significant impact on genome assembly as well as gene finding due to the limitation of short reads used in some assemblies[36]. Short reads may be unable to bridge or cover entire repeat regions within a genome, so it is important to consider the use of long reads from technologies such as Nanopore or PacBio to provide a complete picture of these regions when pursuing a new genome assembly project. It is also possible for repetitive regions to contain genes as well, making for an interesting investigation in regards to *Trichoderma*, as fungal genomes have been shown to contain many repeat regions with a high concentration of A and T nucleotides[38]. Once these repetitive regions have been identified, the genome could be masked to exlude these regions in downstream processing if desired, as these regions may be poorly assembled and may result in found genes that do not truly exist in those regions. However, this may not be as common today, as repetetive regions have been shown to contain genes as well[33]. This may affect the gene finding process described later and may be an interesting topic to look into considering the large number of available gene finding programs.

2.9 Centromere Identification

A centromere is a region of a chromosome that is crucial for the proper cell division. These regions are the main anchor for microtubules, which are a cellular structures used that attach to centromeres to separate chromosomes during both mitosis and meiosis. Centromeres are critical to the survival of an organism, with malfunctions in the process of cell division usually resulting in potential disease and fatal outcomes[27]. Centromeric sequences can be comprised of several different genetic components, with repetititve regions being the most prevalent in the forms of satellite DNA and transposable elements. In addition to centromeric regions, there are flanking pericentric regions with their own properties, including potential candidates for small-interfering RNAs[27]. Identification and consideration of centromeric regions may prove useful when comparing the outputs of gene finding tools, as the underlying properties and structure of the genetic sequence differ in comparison to typical coding regions of DNA.

2.10 Gene Finding Methods

Gene finding (or gene annotation) has been a long standing computational problem in bioinformatics, which concerns itself with identifying potential genes within assemblies based on patterns or pre-existing experimental evidence evidence considered by the gene finding program. This process is critical for unraveling and understanding the complex processes occurring in all forms of life with applications in medical science, agriculture, biomanufacturing, environmental studies and many others. In a general sense, gene finding programs operate by searching for patters or indicators showing that a gene of feature may be present. The most basic indicators being start and stop codons, with introns and exons in between should the sequence match the applied model. The results produced by gene finding tools can vary considerably for a number of reasons, including quality of the assembly, the intrinsic model used by the gene finder, filtering criteria, and even the nature of the organism and assembly itself. Given the broad applications, choice of gene finding tools, and the variability of assemblies being considered, it is important that we gain a deeper understanding of these tools prior to putting them to use.

There are two common methods for gene finding, those methods being *ab initio* methods, where programs search for patterns and gene structures, and similarity or evidence-based searches, which use prior information such as RNAseq data, expressed sequence tags and expressd protein sequences to identify genes within a new genome[12]. Complicating the process more is the introduction of introns and alternative splicing in eukaryotes, making it possible for one gene to have several possible transcripts at the same locus. An example of an *ab initio* method would be GeneMark-ES[20], while an evidence based tool would be Braker2[8]. *Ab initio* gene finders typically predict genes using a Hidden Markov Model (HMM)[12]. These predictions are based on 'signals' or features associated with a gene, such as the usual start, stop, exon and intron portions of a gene as well as upstream promoter sequences and more. In this case, these signals would be considered states

in the terminology associated with HMMs. Gene finders wish to predict these states based on observations, or sequences presented to the model. HMMs in gene finding tools are trained beforehand and then applied to a sequence. This means that a gene finding program may not be trained in the context of any assembly provided to it, and thus may miss genes that are unique to the assembly in question. On the other hand, while still relying on HMMs for a 'base' set of predictions, evidence-based gene finding tools leverage new evidence that may be outside the scope of the pre-existing model[**?**]. As an example, an evidence-based model would be useful in a situation where you are interested in annotating a new assembly for a non-model organism. The addition of experimental data provides context specific to your assembly of interest while still retaining the predictions from existing HMM models.

There are also other aspects of gene finding tools that are important to consider. These include features such as whether or not the gene finders find non-coding RNAs, annotation of 5' and 3' UTR regions, and in the case of ab-initio methods, the assumptions made by the underlying models used for gene finding. These features and others can influence a user's decision on which gene finding tool to consider and will complicate comparative analysis of multiple gene finding tools. (citation needed somewhere in here)

2.11 RefSeq

First, we will briefly discuss the RefSeq annotation process. RefSeq annotation is only applied to data that is submitted to NCBI. The RefSeq Eukaryotic Genome Annotation Pipeline[23] is a genome annotation process developed and maintained by NCBI. The pipeline is not directly publicly available to public users, and requires submission of data to NCBI. Once data is submitted to NCBI, the RefSeq annotation pipeline may be applied upon request only if the genome is the highest quality assembly for the species in question or if the genome is of significant interest to the scientific community, limiting reach of the annotation process to many users. The pipeline supplies existing RNAseq, CDS and protein sequences to NCBI's in-house gene prediction tool Gnomon, which produces trained models for gene prediction. While the tools used for alignment and processing of supporting sequence information are listed, the inner workings of Gnomon are not well documented, at least from the public perspective, and I was unable to find Gnomon in any compilable or executable form during my search. Recreation of the RefSeq pipeline would prove extremely challenging if not impossible without supporting information. Run times for this pipeline are difficult to determine due to the hidden nature of the pipeline, unknown compute resources and varying quantities of data used. The RefSeq annotation process produces comprehensive outputs, including CDS sequences, translated CDS, RNA from genomic sequences, proteins, feature counts and tables, and finally GFF and GTF formatted annotation files for these features.

```
>rna1
ATCGTAGTGCTGTGATCGTAGTCGGATTGGACATGATCGATTCGAT
TGATTGATCGATTGCTAGAAATCGATCGGCTCGTATATCGATCGTA
ATGTCGTTAGTCGAT
>dna2
ATTAGTCGATGCTAGCTGATGGTTAGCTAGTTCGATTCGGTATAGATCTAGGCTTAGAGATATCGCGCTAGCTAGCTAGC
```

Figure 2.1: Example of two FASTA sequence entries. One example with sequence characters split across multiple lines, and one showing all sequence characters on the same line.

2.12 InterProScan

The outputs from gene finding tools are a set of potential genes that fit the model used by each tool. While they are considered genes, the use of the word gene is used in a very loose sense, in that these genes may or may not be functional or match any existing gene sequences from previous research. Typically, to confirm the 'correctness' of predicted genes, the outputs from a given tool are used in a sequence similarity search against a reference set of genes or a large datasbase comprised of multiple organisms. This approach is straightforward, but can introduce bias from database choice and also allows for vague or loose matches, depending on the parameters used and the interpretation of the results. Another approach is to use InterProScan, which is a tool used for functional annotation of proteins using evidence from a variety of databases [**?**]. The presence of some form of functional domain or annotated structure in a predicted gene sequence is reasonable evidence for the existence of a predicted gene. In addition. This approach also avoids the problems associated with similarity-based approaches.

2.13 File Formats

2.13.1 FASTA

One of the most popular formats for sequences of DNA, RNA and amino acids is the FASTA format. The FASTA format consists of one or more entries containing two or more lines. The first line of an entry is the ID line, which must begin with a greater-than ('¿') character, followed by an ID and any other pertinent inormation for the following sequence. The greater-than character is the indicator that a new sequence has begun. The following line(s) contain the actual sequenced nucleotides or amino acids, which can be contained on one line or split across many lines. An example of multiple FASTA entries are shown in figure 2.1.

2.13.2 FASTQ

Another popular sequencing format is the FASTQ format. This format is very similar to the FASTA format but with the addition of two more lines per sequence entry and a change to the character indicating the beginning of a new sequence entry. An example of a FASTQ entry is shown in figure 2.2. In FASTQ formatted entries, the greater-than ('¿') character is swapped with the at ('@') character. The IDs for the

```
@M00833:808:000000000-CJYGN:1:2105:15607:1332 1:N:0:1
CAACTGACGTAGGTGGGACATTCTCCAACGATGATCCTGCGAGGAATGCGATGGAGGGGCGCAGATGCGGGTTTTGGTTCTAGATG
CCCCTGCAGCGGTTCGCATGTGGCTTTGCCTTTTTTTCTTTTTTTTTTTTTTTGTTTTTCTGTTTTTTTAGATGTTCGGAACAGC\
CTCTTTTTGGACTTTTTTTTTTGGAAACTCGCTGCTGGAATGCTGGGTTTTGGTCCCTTTTAGGTTTTGGCTTACTGTT
+
AABCBFFFBAAFGFGGGGGGGGHHHHHGFGGHGFHHHHGHGGGGGBFGHGGHGEHHGGGGGGGGGGHHGGGEFGGHGHHHHHHGGH
HHHGHGHHGHGGGCFGGGGGFHHHHHHHHHHHHGHGGGHHHHHGGGGGFGC-;-;BFFE/00:FFF.-D.B000;FBA-..-//:\
9//;BFFF=-;0;FFFFFFFF;-::B9/00.-...0;0.;009000./.9;.-:0F.F00;0:0009..-00:0;00;0
```

Figure 2.2: Example of the four lines in a FASTQ entry.

```
ctg000000    AUGUSTUS    gene   10842 11309 .     -     .     ID=g4;
ctg000000    AUGUSTUS    mRNA   10842 11309 0.6   -     .     ID=g4.t1;Parent=g4;
ctg000000    AUGUSTUS    stop_codon   10842 10844 .     -     0     ID=g4.t1.stop1;Parent=g4.t1;
ctg000000    AUGUSTUS    CDS    10842 11309 0.6   -     0     ID=g4.t1.CDS1;Parent=g4.t1;
ctg000000    AUGUSTUS    exon   10842 11309 .     -     .     ID=g4.t1.exon1;Parent=g4.t1;
ctg000000    AUGUSTUS    start_codon  11307 11309 .     -     0     ID=g4.t1.start1;Parent=g4.t1;
```

Figure 2.3: An example of GFF entries for a sinlge gene.

sequence also follow a specific format, which provide information about the sequencing run and flowcell that the read was sequenced on. This information can then be traced back to the sequencing experiment in the case that there were errors or anomalies in the output from the experiment. Following the ID is the string of base calls. The third line in a FASTQ entry is a plus ('+') character, which indicates that the sequences character line has finished. Following the plus ('+') character is another sequence of characters, this time indicatintg the quality of basecall for the corresponding nucleotide base calls in the second line. The quality information included in FASTQ files are used to assess the quality of a sequencing run and extensively used in downstream processing steps, most notably in alignments.

2.13.3 General Feature Format - GFF

General feature format (GFF) is a popular format for storing information about features relative to a position on an genetic sequence, and comprises a large portion of annotation results from this work. These features can be whatever the user desires, as long as the feature entry follows the required GFF guidelines. Relative to a reference sequence, each GFF entry contains the following tab-delimited columns: sequence ID, source, feature type, start position, end position, score, strand, phase, and a semi-colon delimited list of attributes. GFF files are widely supported accorss bioinformatics tools, making them highly versatile while also remaining relatively simple in nature but also allowing for storage of more complicated items via the attributes column. One significatn useage of GFF files is in visualization of features against the reference sequence from which they were derived. Most genome viewers (or browsers) support GFF files as input, allowing intuitive visualization of many features when overlayed on a reference sequence. An example of a GFF entry can be seen in figure 2.3.

Background for BUSCO (from research questions) As more and more genomes are assembled for new organisms, a tool was developed to evaulate assemblies and subsequent annotations from the pserpective of gene orthology. As genomes diverge evolutionarily, it is expected that some genes will be conserved. The BUSCO (Benchmarking Universal Single-Copy Orthologs) tool and datasets were developed to assess

completeness of an annotation in comparison to evolutionarily conserved genes.

3 Research Questions

3.1 Research Questions

With an ever-increasing number of gene prediction tools available to users, it is important to assess and understand their behaviour and performance in the context, particularly in the context of new genome assemblies of lesser studied organisms, where a reference prediction set may not be available. The main purpose of this research is to evaluate and compare gene finding tools in the context of *Trichoderma* assemblies where a gold standard set of gene predictions does not exist. To assess behaviour and performance in these contexts, we have defined X problems to profile the selected gene finding tools. In addition to applying selected gene finding tools to novel *Trichoderma* isolates, Tsht20 and DC1, we also applied selected gene finding tools to existing *Trichoderma* assemblies from the National Center for Biotechnology Information (NCBI).

3.2 *Trichoderma* Assembly Results

Since gene finding tools operate on an assembled genomic sequence, it must follow that the results will be influenced by the supplied assembly. Before applying gene finders to the new assemblies, we should first investigate the new assemblies by generating general assembly metrics for the new assemblies to contrast and compare with existing assemblies. We ask: **how do assemblies of DC1 and Tsth20 compare to existing *Trichoderma* assemblies?** With these isolates being from the *Trichoderma* family, we expect assembly metrics to be similar in nature to existing assemblies from NCBI, but do they?

3.3 Profiling of Gene Finding Tools

Different gene finding tools may predict different types of features associated with gene structures. The question arises: **which (if any) gene finders predict additional features outside of the standard gene model?** Additional features in this case include promoter sequences, transcription binding sites, activating sequences and other upstream or downstream sequences. In addition, different gene finding tools employ differing programming languages and algorithms which raises several questions. **How are these gene finding tools implemented? Is the software straightforward to install? Are the tools user-friendly? What is the processing time and memory consumption of different gene finding**

tools in the context of *Trichoderma*?

3.4 Number of Features Predicted

Do gene finders predict similar numbers of features in the context of *Trichoderma* genomes? One common method for evaluating gene finding tools is by looking at the number of features predicted by each tool. These features make an obvious point of comparison for selected gene finding tools. The term 'feature' here is somewhat ambiguous, referring to many possible categories of genomic feature. For each gene finding tool, we compare the counts for predicted genes, transcripts, and coding sequences.

3.5 Lengths of Predicted Genes

Genome assemblies can contain a wide range of gene lengths. For some users, genes of a specific length may be a key point of interest, so the ability of a gene prediction tool to capture the broad range of possible gene lengths is another important metric for comparison. Thus we ask the question: **do different gene finders predict genes of similar lengths in *Trichoderma*?**

3.6 Performance in Regions of Anomalous Sequence Content

One of the inspirations for this research is the unique composition of genomic sequence in *Trichoderma*. Results from the assembly process show that GC content in *Trichoderma* strains is abnormal throughout most assemblies. These regions of assemblies present an interesting opportunity to assess gene finding performance in regions of anomalous GC content. The question follows: **do gene finders behave differently in regions of anomalous sequence content?**

3.7 BUSCO Completeness

The use of existing benchmarks for gene finding performance is useful when assessing performance of gene finding tools, particularly in the case of genes that should be evolutionarily conserved. **Do gene finders predict conserved single-copy orthologs expected in fungal genomes?**

3.8 Identifying Regions of Agreement and Disagreement

With predicted genes from several tools available, the question we would like to ask is whether or not the gene finders agree with one another for any given prediction. To answer this question, we will identify 'regions' of overlapping predictions. A region can be defined as a start and stop position of a set of individual or overlapping features from one or more gene finding tools and external sources. With regions identified,

we can determine agreement, or more importantly, disagreement in predictions between gene finding tools from which we can ask: **do gene finders agree on their predictions? If no, to what extent do they disagree? Are there genomic regions where agreement or disagreement are more prevalent?**

3.9 Validation of Predicted Genes via InterProScan

In an effort to validate, or at least provide supporting evidence for any given gene prediction we will apply InterProScan to coding sequences predicted by each of the gene finders to identify features associated with protein function. Genes will be considered as 'valid' if the gene's protein sequence contains binding sites, motifs, or other fucntional characteristics of proteins. Using the results from InterProScan, we ask the question: **in the context of *Trichoderma*, do proteins predicted by gene finders contain functional signatures?**

3.10 Identification of a Core *Trichoderma* Gene Set.

Using results from identification of regions in section 3.8 we will identify a shared or 'consensus' set of predictions for each *Trichoderma* assembly. **Can we identify a set of consensus predictions from gene finding results? What are the properties of the resulting set?**

3.11 Selection of a Gene Finding Tool

With all the results generated, we can provide insight to the question: **which gene finding tool should one choose?**

4 Data and Methodology

4.1 Workflow Overview

The general methodology for this work is described in figure 4.1. Technical portions of figure 4.1 are discussed in the following sections.

4.2 Sequence Pre-processing and Assembly

Genomic sequences were generated for DC1 and Tsth20 using Nanopore[37] and Illumina[5] sequencing technologies. Nanopore data was not processed prior to assembly. Illumina sequencing data was filtered using Trimmomatic v0.38[6] with filtering criteria as follows: SLIDINGWINDOW:4:28, LEADING:28, TRAILING:28, MINLEN:75. Only surviving paired-end reads were used for further analysis.

Genomic Nanopore sequences from DC1 and Tsth20 were assembled in to contigs using NextDenovo[17] v2.5.0 and then polished with Illumina sequences using NextPolish[16] v1.4.1. Assembly and polishing were both performed using default parameters. Final assembly metrics were calculated using QUAST v5.0.2[15] with default parameters.

4.3 Gene Prediction Using GeneMark-ES

Ab initio gene finding was performed using GeneMark-ES v4.71[7]. Default parameters were used in all cases except for the fungal option, which was allowed the use of a GeneMark model specific to fungal genomes.

4.4 Gene Prediction Using Braker2

For evidence-based gene finding, Braker2 v3.0.2[8] was used with *Trichoderma reesei* selected as the reference organism for training. RNAseq datasets from *T. reesei* were downloaded from the NCBI short-read archive using the sra-toolkit[24] and were not trimmed or filtered prior to their use in Braker2 training. Default parameters were used to train a Braker2 model on *Trichoderma reesei* data except in the case of the fungal option, which was used for this analysis for improved gene-finding performance in fungi. The trained Braker2 *T. reesei* prediction model was then applied to all assemblies with default parameters.

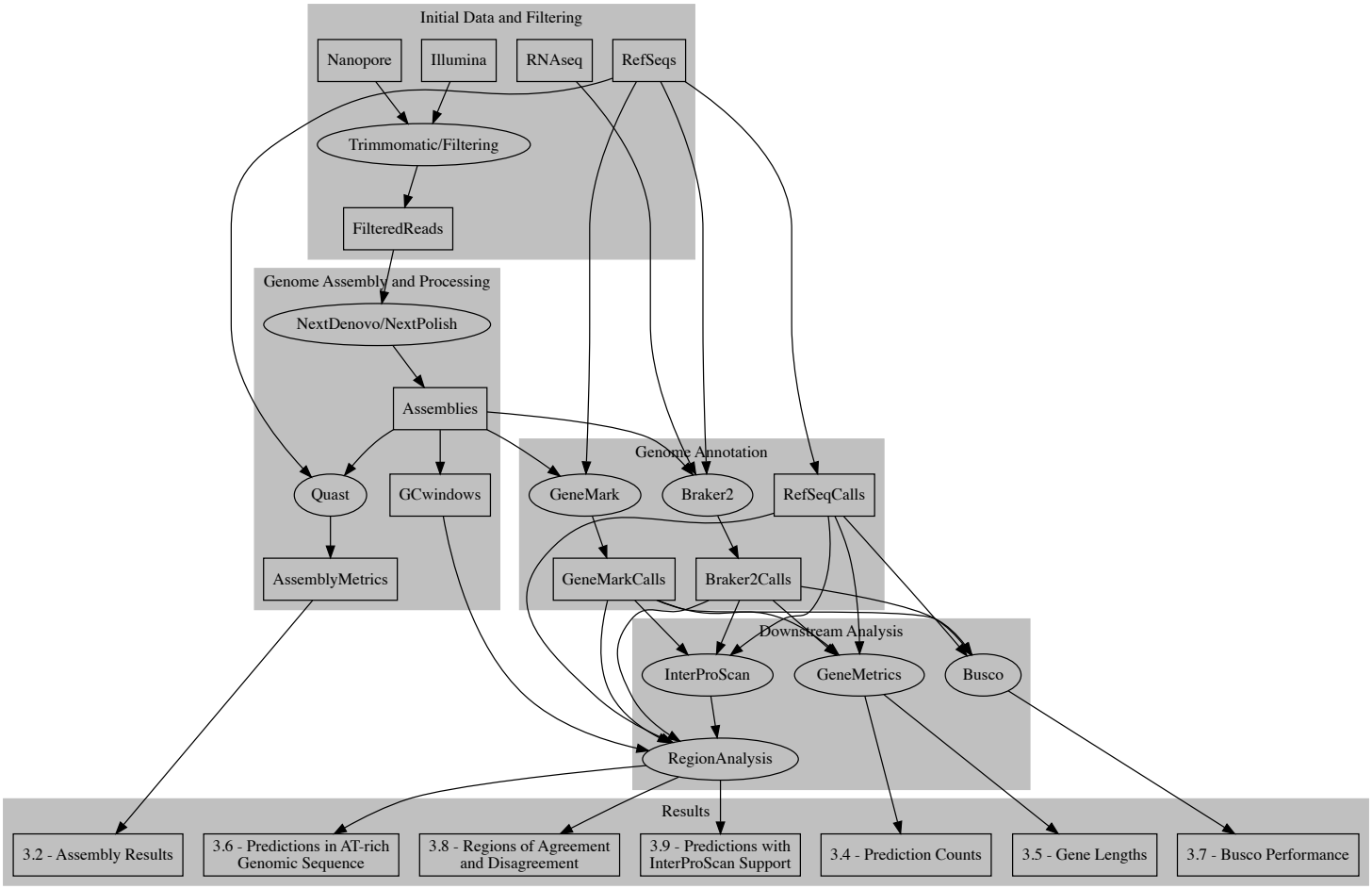


Figure 4.1: A flowchart of the methodology followed in this research. The workflow is broken up into sections based on the processes and data involved at each step. Oval-shaped nodes represent steps involved in the processing of data, while rectangular nodes represent intermediate datasets and results.

Braker2 also provides a UTR option to predict upstream sequence features, but that option is experimental and was left off for this work.

4.5 Binomial Tests on CDS Lengths

In section 5.4, the coding DNA sequences (CDS) of predicted genes were tested using a binomial test, under the null hypothesis that the probability of a gene being in normal or AT-rich genomic sequence is proportional to the total lengths of normal or AT-rich sequence in the assembly.

4.6 Evaluating Gene-Finder Performance Using BUSCO

To assess the general performance of gene-finders, BUSCO v5.7.1[32] was selected as a tool to determine a gene-finder's ability to predict a set of conserved ortholagous genes. The BUSCO Metaeuk pipeline was run using the BUSCO fungi_Odb10 fungal lineage dataset to capture highly conserved genes in fungal genomes. This pipeline was applied to all genome assemblies used in this work. (note for us to discuss: I used transcriptome mode...)

4.7 Region Identification Procedure

Predictions and results from previous steps were then processed into overlapping regions of predictions and annotations. A region is defined as a set of start and stop genomic coordinates which contains one or more overlapping feature. We also define a feature as a set of start and stop positions associated with a single gene prediction, annotation or other data point of interest.

Overlapping features were processed into regions using a concept similar to Allen's interval algebra[11]. The algorithm first sorts all features on each contig by start coordinate and then iterates over each feature, identifying overlaps with previous features to identify regions of overlapping features. Pseudocode for the algorithm is shown in algorithm 1. This processing was performed with Python v3.12.4[13] and the GFF package from BCBio[9] for easy GFF parsing. The resulting regions are then further processed to identify trends and behaviours of gene finders in the context of other annotated information and features.

4.8 Functional Annotation Using InterProScan

The predicted proteins from each gene finder were functionally annotated using InterProScan v5.65-97.0[18] without the match lookup service and with the following analyses included: AntiFam-7.0, CDD-3.20, Coils-2.2.1, FunFam-4.3.0, Gene3D-4.3.0, Hamap-2023_01, MobiDBLite-2.0, NCBIfam-13.0, PANTHER-18.0, Pfam-36.0, PIRSF-3.10, PIRSR-2023_05, PRINTS-42.0, ProSitePatterns-2022_05, ProSiteProfiles-2022_05, SFLD-4, SMART-9.0, SUPERFAMILY-1.75. InterProScan was run on all predicted proteins using default

Algorithm 1 the general algorithm underlying the region identification process.

INPUT: list of GFF files, $GFFList$

RETURNS: list of regions, $regionList$

for `<each GFF in GFFList>` **do**

    $allFeatures \leftarrow GFF.features$

end for

$sortedFeatures \leftarrow sortStartCoord(allFeatures)$

$regionList \leftarrow []$

for `<each feature in sortedFeatures>` **do**

    if $feature = firstFeature$ **then**

        $currentRegion \leftarrow newRegion$

        $currentRegion.start \leftarrow feature.start$

        $currentRegion.end \leftarrow feature.end$

    else if $feature = lastFeature$ **then**

        $regionList \leftarrow region$

        $return\ regionList$

    else if $feature\ overlaps\ currentRegion$ **then**

        $currentRegion \leftarrow updateStartStopPositions$

        $currentRegion \leftarrow feature$

    else

        $regionList \leftarrow currentRegion$

        $currentRegion \leftarrow newRegion$

        $currentRegion \leftarrow feature.start$

        $currentRegion \leftarrow feature.end$

    end if

end for

$return\ regionList$

parameters. The resulting protein annotations were mapped back to the gene predictions and processed using the region identification algorithm in section 4.7.

4.9 Identification of AT-rich Genomic Windows

Sliding windows of GC content over each *Trichoderma* assembly were calculated using isochore from the EMBOSS tool suite v6.6.0[29]. The sliding window used was 250bp wide with a 50bp shift. A window was classified as AT-rich if the sequence contained lower than 28% GC content. The genomic sequences from AT-rich windows were mapped to a GFF file and processed using the region identification algorithm described in section 4.7.

4.10 Ground-truthing with BLAST

Need to flesh this out if I write results section.

4.11 Visualization of Identified Regions

Results from the region identification process were visualized using the IGV genome browser v2.19.1[30].

5 Results

5.1 Assemblies of DC1 and Tsth20

For general assembly metrics of DC1 and Tsth20, the QUAST[15] tool was used. Results from QUAST are shown in Table 5.1, from which we can make several observations. In DC1 and Tsth20, the total contig counts are an order of magnitude smaller when compared to the other NCBI RefSeq assemblies, indicating highly contiguous assemblies from nextDenovo[17] and nextPolish[16]. This is likely due to the use of Nanopore sequencing in the assemblies of DC1 and Tsth20. The total assembled lengths of DC1 and Tsth20 are similar when compared to RefSeq assemblies, ranging from 38Mb to 42Mb, except in the case of *T. reesei*, which is known to have a significantly smaller genome length[19], at roughly 33Mb.

The largest contig size for each assembly varies greatly. DC1 and Tsth20 have the largest contigs of all assemblies being considered, which is again likely due to the inclusion of long-read sequencing data in the assembly process. The N50 values for all assemblies are above 1Mb, with DC1 and Tsth20 N50s being at minimum two times larger than others assemblies. N50 lengths nearing chromosomal lengths indicates that assemblies of DC1 and Tsth20 are better assembled than the other *Trichoderma* assemblies. While chromosome-scale contigs are not the sole indicator of genome quality, it does provide confidence in the quality of the input data and resulting assemblies. In general, the assemblies of DC1 and Tsth20 are of similar length to existing *Trichoderma* assemblies and the number of contigs reported match the number of 'chromosome' scale contigs reported in other work[19].

During initial investigation of the inputs to the assembly process, we observed that the Illumina reads have a bimodal distributiion of GC content as shown in Figure5.1. To see if this observation extended to the

Strain	Total Contigs	Total Length	Largest Contig	GC%	N50	L50
DC1	8	38.6 Mb	11.49 Mb	47.97	5.69 Mb	3
Tsth20	7	41.58 Mb	8.02 Mb	47.33	6.52 Mb	3
T. harzianum	532	40.98 Mb	4.08 Mb	47.61	2.41 Mb	7
T. virens	93	39.02 Mb	3.45 Mb	49.25	1.83 Mb	8
T. reesei	77	33.39 Mb	3.75 Mb	52.82	1.21 Mb	9

Table 5.1: General assembly metrics produced by QUAST[15] (a genome quality assement tool).

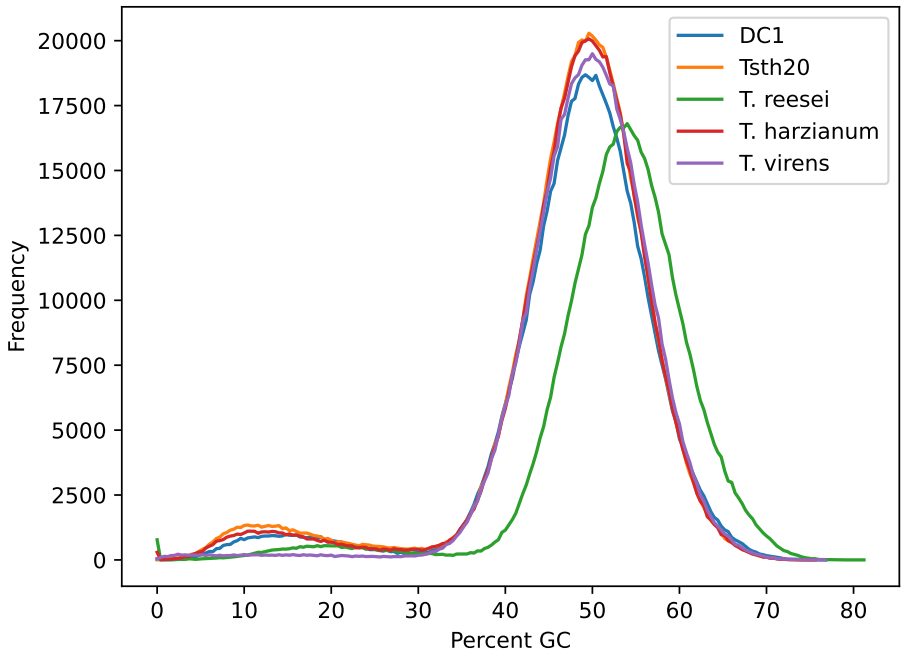


Figure 5.1: Plots showing the frequency of GC values calculated from sliding windows for each assembly.

assemblies as well, 250 bp sliding windows were used to calculate GC content for all assemblies included in this analysis. The results of this analysis are shown in Figure 5.1. AT-rich sequences are visualized on the left peak of the distributions sequences containing 90 percent AT content. Of the included assemblies, AT-rich sequences were identified in DC1, Tsth20, *T. reesei* and *T. harzianum*, with *T. virens* deviating from the other assemblies showing very few AT-rich windows. This stark difference in *T. virens* nucleotide composition may indicated potential assembly issues or even misclassification of the organism. In addition to the confirmation of increased AT-rich sequence content in most assemblies, it appears that the distribution of GC content in *T. reesei* differs from the other assemblies. The curve of GC content for *T. reesei*, visualized in green in Figure 5.1, lies to the right, indicating fewer AT-rich windows and more balanced nucleotide composition in its assembly. While the left tail of the curve also shows an increase in AT-rich sequence composition, with it's peak located farther right on the X-axis than other *Trichoderma* assemblies. Investigation of these AT-rich sequences is continued in section 5.8, where only sequences containing less than greater than 72% AT nucleotide content are considered, as the distributions begin to deviate from the normal distribution at that point.

5.2 Installation and Profiling Gene Finding Tools

While Braker2, GeneMark and RefSeq all provide lists of possible genes for a provided input, the implementation of each tool is different, requiring more or less effort to install and run thn other tools. This section discusses the implementations, installation procedures and execution of GeneMark and Braker2 in more detail. For more information on the versions and parameters used in this processing, please see section 4.4. The computing platform used in this research is hosted and managed by University of Saskatchewan services, which is modelled around the HPC platform used by the Digital Research Alliance of Canada and software is managed similarily.

GeneMark[20] is a gene finding tool developed by the Georgia Institute of Technology with packages prepared for Linux and MacOS. It is provided as licensed product in the form of a package which can be downloaded from their website after submitting a form. Once the necessary information is submitted, the user is provided with a key that must be placed in the appropriate location once the software is downloaded and unpacked. The core controlling methods of GeneMark are written in Perl, accompanied by several Python scripts and compiled exectuables. GeneMark was tested by the developers with Perl version 5.10, and Python 3.3. A number of Perl dependencies are also required, which can be installed via CPAN. The user will have to know which implementation of GeneMark they are wanting to use for their application, as GeneMark has several variations it can run depending on the desired application. In this work, the GeneMark-ES variant of GeneMark was executed, as it is the self-training *ab initio* GeneMark method for eukaryotic organisms. Options required by GeneMark at runtime are documented in the help message, and simple enough that any user with familiarity of bioinformatics tools should be able to run GeneMark, although documentation for use is only provided by the help message when running the program and not online. In regards to run-time, running the GeneMark-ES pipeline on DC1 with 56 threads finished in 16 minutes. GeneMark's outputs consist of a GTF or GFF file of predicted genes as well as a number of other outputs related to the run.

Braker2[8] is hosted on GitHub as a repository that receives relatively frequent updates with Braker3 being released while working before the completion of this work. Braker is maintained by Katharina Hoff from the University of Greifswald and is available under the Open Source Artistic License. Installation of the repository is a straightforward pull from GitHub. As with GeneMark, Braker2 uses a combination of Perl, Python and other exectuables in its regular use. Downloading the repository itself is not enough for execution, as Braker2 relies on a number of dependencies and bioinformatics tools including Perl and (Perl dependencies), Augustus[34], BamTools[4], BedTools[28], GeneMark[7], StringTie[26], GFFRead[25] and several others. Manual installation of these dependencies would be difficult, time consuming and in general advised against. In this case, many of Braker2's requirements are satisfied by modules already included in the environment, making installation relatively simple if one is familiar with the Digital Research Alliance of Canada's software stack. This case still required installation of some Perl modules in addition to loading necessary modules. Alternatively, one could use a package manager like Anaconda or Minoconda

to handle installation of packages. This is perfectly reasonable, but also requires knowledge specific to Anaconda, which can be complicated and frustrating for users with little software management experience. Once installation is finished, the Braker2 pipeline is relatively straightforwad to run as well, with excellent documentation included both online and through the built in help message. The training, including RNAseq alignments, and gene finding pipeline in this case took 1 hour and 17 minutes using 60 threads. Applying the Braker2 trained gene finding model to DC1 with 60 threads took 21 minutes to complete. Run times will of course vary depending on processing power available to the end user, but in the case of *Trichoderma* genomes, users can expect quick results with relatively little computing power. Once annotation is complete, Braker2 produces a GFF file containing predicted genes along with CDS sequences and amino acid sequences their protein products.

In summation, Braker2 and GeneMark are not direct plug-and-play software packages. Users should expect to encounter issues when getting these programs running in addition to normal downloading and unpacking of software packages so some expertise is recommended. Neither Braker2 nor GeneMark require users to compile software, however Braker2's dependencies may require additional compilation and attention. Once installed, both tools are relatively simple to use with documention available for both on the commandline and excellent documentation available for Braker2 on their GitHub page. Outputs from both tools are similar although Braker2 has the ability to output coding and amino acid sequences for downstream processing. Both tools run in reasonable amounts of time, where in the case of smaller genomes such as *Trichoderma*, users can expect results within a few hours to a day depending on number of computing cycles available to them.

5.3 Initial Gene Finding Results

Counts of genes predicted by Braker2, GeneMark and RefSeq are shown in Table 5.2. Immediately we see that Braker2 predicts far fewer genes in all assemblies, except in the case of *Trichoderma reesei.* This is possibly due to the effects of training the Braker2 gene model using data from *Trichoderma reesei*, which has a significantly smaller genome in comparison to other *Trichoderma* assemblies, although genome size is not always indicative of gene content. Regardless, we observe a difference in the number of genes predicted by Braker2 in comparison to GeneMark and RefSeq. The number of genes predicted by GeneMark and RefSeq are similar, except in the case of *T. harzianum*, in which RefSeq predicts roughly 17% more genes than GeneMark. Braker2 consistently predicts more transcripts than GeneMark and RefSeq. RefSeq also appears to predict multiple transcripts for each gene but in fewer numbers than Braker2. Transcript prediction counts in *T. harzianum* from RefSeq are also interesting, with RefSeq predicting fewer transcripts than genes. Why this is occurs is unknown but may warrant further investigation.

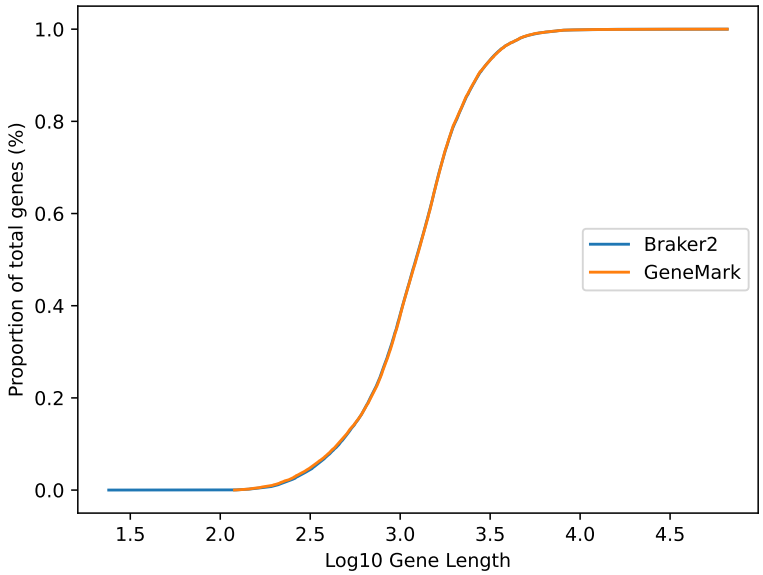
Assembly	Braker2		GeneMark		RefSeq	
	Genes	Transcripts	Genes	Transcripts	Genes	Transcripts
DC1	8546	8637	11353	11353	N/A	N/A
Tsth20	8784	8858	12362	12362	N/A	N/A
T. reesei	9659	10175	9196	9196	9109	9118
T. harzianum	8314	8385	12164	12164	14269	14090
T. virens	7801	7863	11866	11866	12405	12406

Table 5.2: Number of genes and transcripts (isoforms) predicted by each gene finder for each *Trichoderma* genome.

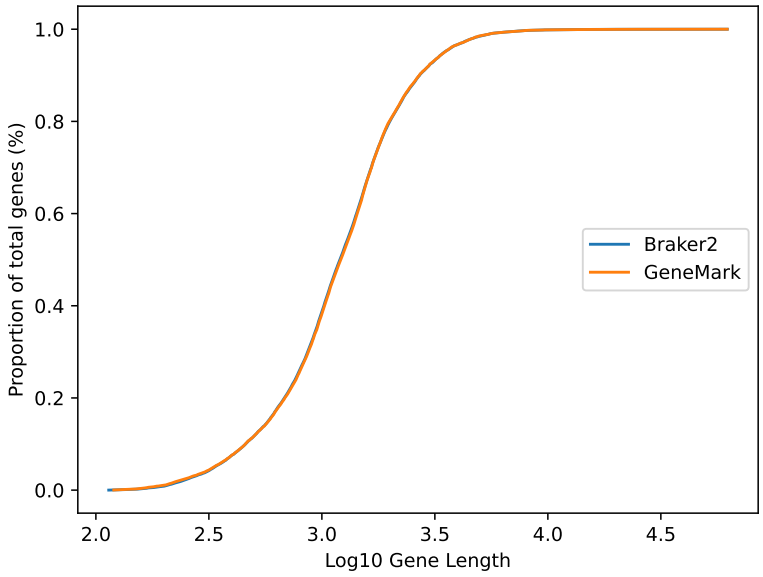
5.4 Distribution of Predicted Coding Sequence Lengths

To better understand and compare the distributions of CDS sequences predicted by Braker2, GeneMark, and RefSeq, the cumulative distribution function for the lengths of CDS sequences for each gene finding tool are shown in Figures 5.2, 5.3 and 5.4. The $\log_{10}$ values of gene lengths were used as the abscissa for a better visulazation of the distributions. In DC1, the curves from Braker2 and GeneMark follow each other closely, with the only variation being genes of short length, where Braker2's curve extends beyond that of GeneMark, indicating that Braker2 predicts the shortest genes in all assemblies except *T. virens*. In the case of Tsth20, the curves are nearly identical. In *T. reesei*, we see disagreement in the curves for shorter genes, with Braker2 appearing to predict a larger fraction of shorter genes than GeneMark and RefSeq. The right sides of the curves trend toward similar predicted gene lengths. In *T. harzianum*, RefSeq deviates from GeneMark and Braker2, predicting more genes of short length, while Braker and GeneMark appear to be in near-complete agreement except in the case of very short genes. Finally, in *T. virens*, we see the RefSeq curve predicting a larger fraction of shorter genes once again, although the deviation is not as drastic as in *T. harzianum*.From these plots, we can say that visually, gene finding tools appear to predict different lengths of genes. We also observe that Braker2 typically predicts the shortest genes of all three gene finding methods.

To confirm that CDF curves differ, two-sided two-sample Kolmogorov-Smirnov[1] tests were performed using the $\log_{10}$ transformed gene lengths, with the null hypothesis being that frequency of genes predicted in AT-rich and normal genomic sequence are same. Results are presented in Table 5.3. In the cases of DC1 and Tsth20, we see that in agreement with Figure 5.2, Braker2 and GeneMark do not produce statistically different lengths of genes, which is interesting considering that the Braker2 includes experimental evidence for another assembly while GeneMark does not. In *T. reesei*, Braker2's predicted gene lengths are significantly different from both RefSeq and GeneMark, which is also evident in the CDF plots. The same cannot be said for *T. harzianum* and *T. virens*, where RefSeq is significantly different from both GeneMark and Braker2, which are not significantly different from each other. It is also notable that RefSeq and GeneMark predict similar gene lengths in *T. reesei*, but not in *T. harzianum* and *T. virens.*

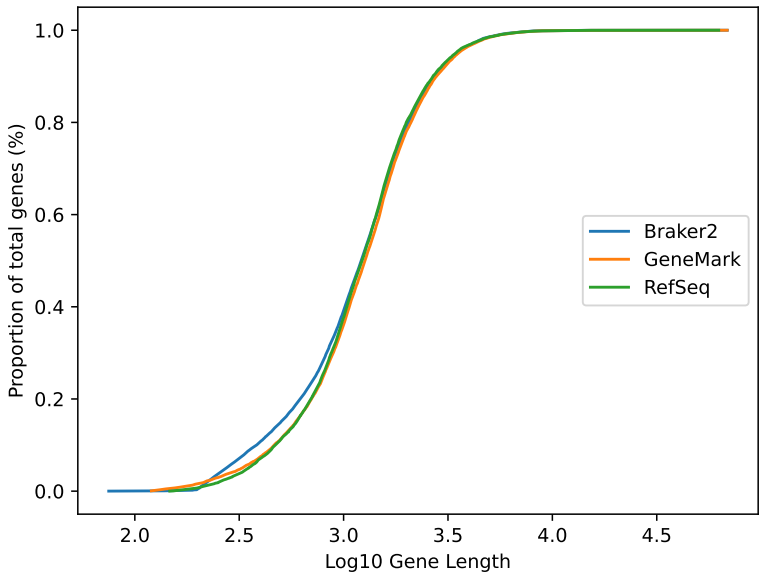


(a) DC1

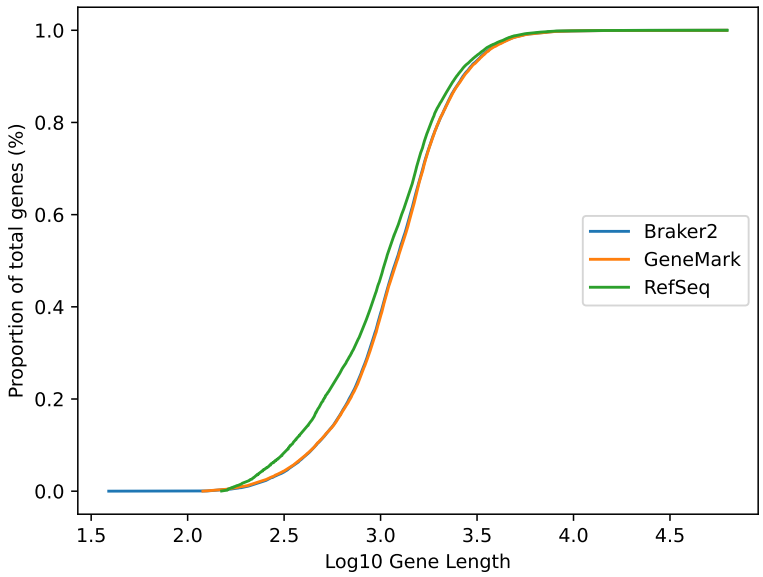


(b) Tsth20

Figure 5.2: Plots of the cumulative distribution function for CDS lengths produced be each gene finding tool applied to DC1 and Tsth20

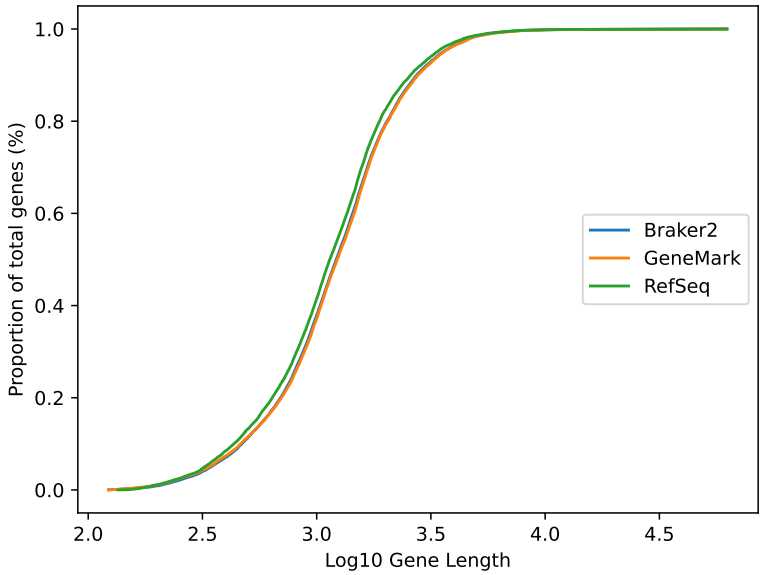


(a) *T. reesei*



(b) *T. harzianum*

Figure 5.3: Plots of the cumulative distribution function for CDS lengths produced be each gene finding tool applied to *T. reesei* (GCF_000167675.1_v2.0) and *T. harzianum.* (GCF_003025095.1_Triha_v1.0)



(a) *T. virens*

Figure 5.4: Plots of the cumulative distribution function for CDS lengths predicted by each gene finding tool when applide to *T. virens* (GCF_000170995.1_TRIVI_v2.0).

Genome	Tool #1	Tool #2	*P*-value
DC1	Braker2	GeneMark	0.999
Tsth20	Braker2	GeneMark	0.965
T. reesei	Braker2	GeneMark	$9.481 * 10^{-07}$
T. reesei	GeneMark	RefSeq	0.002
T. reesei	Braker2	RefSeq	$1.340 * 10^{-07}$
T. harzianum	Braker2	GeneMark	0.863
T. harzianum	GeneMark	RefSeq	$4.313 * 10^{-52}$
T. harzianum	Braker2	RefSeq	$4.674 * 10^{-55}$
T. virens	Braker2	GeneMark	0.635
T. virens	GeneMark	RefSeq	$7.352 * 10^{-12}$
T. virens	Braker2	RefSeq	$1.794 * 10^{-09}$

Table 5.3: Table of *P*-values from two-sided two-sample Kolmogorov-Smirnov tests between gene finding tools.

Reference	Ref. Proteins	DC1	Tsth20	*T. reesei*	*T. harzianum*	*T. virens*
Trichoderma atroviride	11807	11552	11080	10601	11081	11078
Fusarium granminarium	13312	10327	10429	10064	10434	10490
Saccharomyces cerevisiae	6014	3537	3517	3445	3509	3500

Table 5.4: tblastn hits from reference protein sequences to selected *Trichoderma* assemblies. Hits are reported if the alignment length is greater than 30% of the reference protein length and if 30% of the aligment length has identical matches.

It can clearly be stated that these gene finding tools predict different distrbutions of gene lengths, paritcularly in *T. reesei, T. harzianum* and *T. virens*. Why that may be the case is difficult to answer without deeper invesitgation. There is clearly an underlying difference between RefSeq and the other gene finding tools. Braker2 and GeneMark tend to be in agreement, except in the case of *T. reesei*, for which Braker2 was specifically trained. We observe that when Braker2 is applied to an assembly from which the training data originated, Braker2 predicts a larger fraction of shorter genes. Conversly, we observe that when Braker2 is trained with experimental evidence and applied to a *Trichoderma* assembly from which the training data did not originate, the distributions of predicted genes lengths do not significantly differ from the *ab initio* gene finder GeneMark.

5.5 BLAST Results

As a control, proteins from three related organisms were used as queries to tblastn[3] on each *Trichoderma* assembly. The organisms and their associated NCBI accession IDs are as follows: *T. atroviride* - GCF_000171015.1, *F. graminarium* - GCF_000240135.3, *S. cerevisiae* - GCF_000146045.2. Results from the tblastn runs are presented in table 5.4. Initial tblastn results appear promising for both the *T. atroviride* and *Fusarium* datasets. All assemblies considered contain at minimum 89% of the reference protein sequences in the case of *T. atroviride* and a minimum of 75% in the case of *Fusarium*. In the case of *S. cerevisiae*, a minimum of 57% of reference proteins matched. It appears that the percentage of hits decreases as evolutionary distance increases. These results provide rough validation that the assemblies contain potential for protein coding sequences.

While successful tblastn hits from input proteins are useful, it is worthwhile exploring those hits in the context of regions and gene predictions within them. Counts of genes from regions with tblastn hits are shown in Table 5.5. In a similar trend to Table 5.4, genes in regions with tblastn hits decrease as evolutionary distance increases. In DC1 and Tsth20, we observe that Braker2 predicts more genes in regions with tblastn hits than GeneMark. RefSeq is not included for DC1 and Tsth20, as there is no RefSeq annotation available. In *T. harzianum* and *T. virens*, Braker2 and RefSeq appear to have similar counts of genes in regions with tblastn hits. Interestingly, in the case of *T. reesei*, there are more genes from GeneMark in regions with tblastn hits, which is not the case in *T. harzianum* and *T. virens*.

Subject	Query	Braker2	GeneMark	RefSeq
DC1	*T. atroviride*	5902	4679	N/A
DC1	*F. graminarium*	4955	4114	N/A
DC1	*S. cerevisiae*	2105	1850	N/A
Tsth20	*T. atroviride*	6065	4626	N/A
Tsth20	*F. graminarium*	5365	4191	N/A
Tsth20	*S. cerevisiae*	2211	1869	N/A
T. reesei	*T. atroviride*	5072	5174	4989
T. reesei	*F. graminarium*	4577	4685	4529
T. reesei	*S. cerevisiae*	2055	2114	2022
T. harzianum	*T. atroviride*	6363	4611	6835
T. harzianum	*F. graminarium*	5659	4198	5982
T. harzianum	*S. cerevisiae*	2424	1963	2560
T. virens	*T. atroviride*	6256	4437	6415
T. virens	*F. graminarium*	5568	4075	5664
T. virens	*S. cerevisiae*	2318	1861	2352

Table 5.5: Counts of transcripts predicted by Braker2, GeneMark and RefSeq in regions containing tblastn hits from *Ttrichoderma atroviride*, *Fusarium granminarium*, and *Saccharomyces cerevisiae*.

Another interesting observation is comparing the coverage, or completeness, of the input and query sequence. As mentioned, it appears that high proportions of query sequences are being mapped to the subject, but the gene predictions are not as well represented in regions where those tblastn hits occur. A likely reason for this is innapropriate alignment parameters. More flexible gap parameters and the use of a different scoring matrix would likely produce better results, but optimization of parameters was not performed in this analysis. It is also possible that the post-alignment filtering criteria were inappropriate, resulting in short alignments that are true in nature, but may not be indicative of the presence of a truly similar protein.

Overall, it appears that using tblastn with proteins from related organisms may be a viable option as a control when no reference or model organism is available. At the very minimum, tblastn hits can be used as another piece of supporting evidence for the existence of a gene, with one caveat being imperfect coverage of predictions. The issue of imperfect coverage may be mitigated by parameter optimization. It is also possible that one could employ a similarity-based cutoff on the proportion of predicted genes in regions with tblastn hits as an indicator of gene finding performance, however this would require further analysis, preferably with more closely related organisms.

Strain	Complete	Single	Duplicated	Fragmented	Missing
DC1	99.5	80.2	19.3	0.1	0.4
Tsth20	99.9	81.7	18.2	0.0	0.1
T. harzianum	99.7	80.2	19.5	0.0	0.3
T. virens	99.8	79.0	20.8	0.1	0.1
T. reesei	99.9	85.5	14.4	0.1	0.0

(a) Braker2

Strain	Complete	Single	Duplicated	Fragmented	Missing
DC1	99.2	98.8	0.4	0.3	0.5
Tsth20	99.8	99.1	0.7	0.0	0.2
T. harzianum	99.6	98.9	0.7	0.0	0.4
T. virens	99.7	99.2	0.5	0.1	0.2
T. reesei	99.6	99.5	0.1	0.0	0.4

(b) GeneMark

Strain	Complete	Single	Duplicated	Fragmented	Missing
T. harzianum	99.9	99.2	0.7	0.0	0.1
T. virens	99.5	98.8	0.7	0.3	0.2
T. reesei	99.8	99.5	0.3	0.0	0.2

(c) RefSeq

Table 5.6: Results from BUSCO using the fungal analysis option organized by gene finding tool. The selected BUSCO dataste contains 758 markers. For more information on the categories assigned by BUSCO, please refer to the documentation.

5.6 BUSCO Results

The results of BUSCO analysis using the fungal subset provided by BUSCO are presented in Table 5.6. Results from BUSCO indicate that all gene sets considered in this analysis have a BUSCO completeness of 99.2% or higher, with a maximum completeness of 99.9% for some gene sets. In general, Braker2 and RefSeq have the most BUSCO complete sets of gene predictions of the three tools considered. Interestingly, Braker2 produces far more duplicated BUSCO matches than both GeneMark and RefSeq. Examining the BUSCO output logs, this appears to be due to Braker2 predictin more than one transcript for some genes predictions, resulting in multiple similar proteins. In general, all gene finders perform exceptionally well in regards to BUSCO performance. While these results do not capture the entire set of genes possibly present in these *Trichoderma* assemblies, they do confirm that the gene finders are at minimum predicting many evolutionarily conserved fungal genes.

Tool	BUSCO ID	Annotation	DC1	Tsth20	*T. reesei*	*T. harzianum*	*T. virens*
Braker2	195619at4751	Pyridoxal phosphate-dependent transferase		✓	✓	✓	✓
Braker2	285254at4751	Aminoacyl-tRNA synthetase	✓	✓	✓		✓
Braker2	348020at4751	Formyl transferase			✓		✓
Braker2	497024at4751	Zinc finger C2H2-type		✓	✓	✓	✓
GeneMark	195619at4751	Pyridoxal phosphate-dependent transferase		✓	✓	✓	✓
GeneMark	285254at4751	Aminoacyl-tRNA synthetase	✓	✓	✓		✓
GeneMark	348020at4751	Formyl transferase					✓
GeneMark	438731at4751	LSM domain	✓	✓		✓	✓
GeneMark	470813at4751	Ubiquitin-conjugating enzyme					
GeneMark	497024at4751	Zinc finger C2H2-type		✓	✓	✓	✓
RefSeq	494at4751	Midasin	N/A	N/A			✓
RefSeq	315802at4751	tRNA dimethylallyltransferase	N/A	N/A	✓	✓	
RefSeq	352224at4751	YEATS	N/A	N/A		✓	

Table 5.7: The presence (✓) or absence of all BUSCO IDs missed by Braker2, GeneMark and RefSeq in each *Trichoderma* assembly.

While BUSCO matches are a good metric for general performance of gene finders, it is also important to investigate BUSCO proteins without matching gene predictions. Tables **??**, 5.7 and **??**, show breakdowns of genes missed by each gene finder across the *Trichoderma* assemblies. Braker2 misses four unique proteins across the five *Trichoderma* assemblies, with only one protein missing in more than one assembly. This protein represents a formyl transferase protein. GeneMark predictions miss six unique BUSCO proteins, with two proteins missing in more than one assembly. These proteins are the same formyl transferase missed by Braker2, which was missed in four of the five assemblies, and a ubiquitin-conjugating enzyme, which was missed in all five assemblies. RefSeq, being the gene finder with the most BUSCO complete set of gene predictions, misses only three unique BUSCO proteins in the three assemblies. Those missing proteins represent a YEATS protein domain and Midasin protein. Of those three proteins, the Midasin and YEATS proteins are missed in two of the three assemblies from NCBI. Those missing proteins represent a YEATS protein domain and Midasin protein.

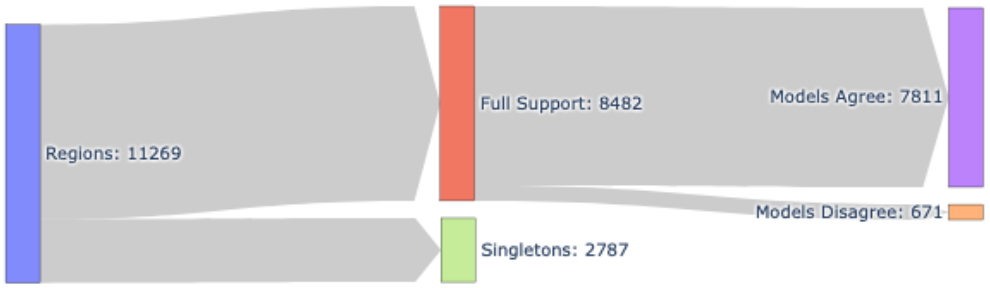
Braker2, GeneMark and RefSeq all demonstrate excellent coverage of the BUSCO fungal protein set, indicating that these gene finders are capable of predicting genes that are expected to be present in these assemblies. From this we can say that the foundations of the underlying gene models used by each gene finder

are solid. Braker2 produces more duplicate matches than GeneMark and RefSeq, but this is likely due to mutliple isoforms of possible genes being present in the input data. Despite excellent coverage of the BUSCO fungal proteins, all three gene finders miss some BUSCO proteins in their predictions. GeneMark misses the most proteins and in paritcularly struggles with predicting a formyl transferase and a ubiquitin-conjugating enzyme. Braker2 also appears to have difficulty predicting a formyl transferase just as GeneMark did. RefSeq misses the fewest BUSCO proteins and does not appear to systematically miss certain proteins, although it is hard to draw a conclusion with only three assemblies considered. It is also worth noting that RefSeq misses completely different proteins than the other gene finders while Braker2 and GeneMark do share some missed proteins. Finally, we note the possibility that human curation of RefSeq datasets is responsible for these differences, but this requires further investigation.

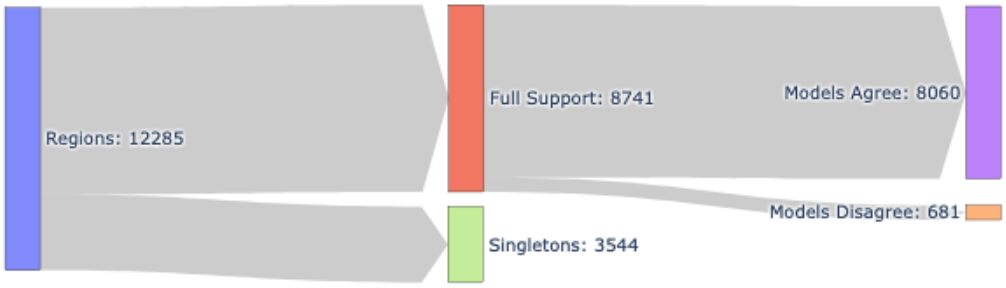
5.7 Region Identification

Visual breakdowns of the types of regions identified and their counts for each *Trichoderma* assembly are shown in Figure 5.5. We will discuss the types of regions, beginning with fully supported regions. These regions, labelled 'Full Support', are sections of genomic sequence where all three gene finders agree that a gene is present in some form. These fully supported regions are then broken down into two categories, based on whether or not the models from each gene finder agree on the start and/or stop positions of the gene model. Regions that have support from more than one gene finder, but not all, are labelled regions with 'Partial Support'. These regions are also broken down in to two subtypes based on whether or not the gene predictions agree on the start and/or stop positions of the gene. Regions with support from only one gene finder are labelled as singletons.

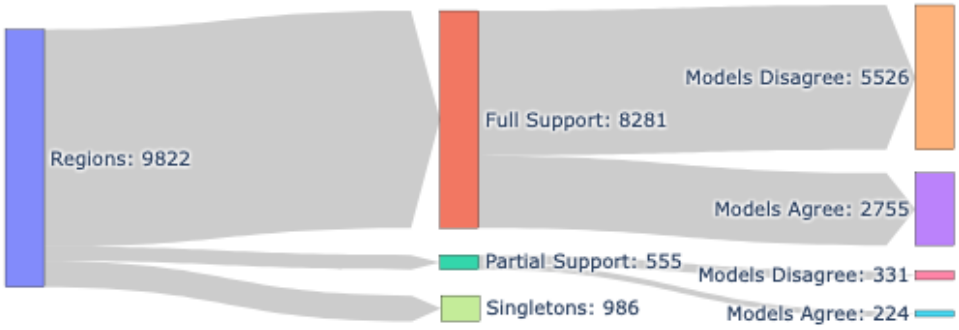
Looking at DC1 and Tsth20 in Figure 5.5, it appears that Braker2 and GeneMark predict genes in the same regions in the 69% and 65% of cases, respectively. For regions with full support in DC1 and Tsth20, the models also tend to agree on the start and stop positions of the gene in that region. This is not generally the case as we will see later when RefSeq is also considered. There are no regions with partial support for DC1 and Tsth20, as there are only two gene finders considered. *Trichoderma reesei* contains the fewest number of regions, which seems to scale appropriately with the total number of predicted genes and assembly length. Eighty-four perfecnt of the regions are fully or partially supported by Braker2, GeneMark and RefSeq, with only 10 percent of the regions being single tool predictions. The most interesting observation here is the extent to which fully supported regions disagree on start and stop positions of the gene(s) in each region. There is clearly a difference in the gene models being produced by these gene finders. If there is also more disagreement on the start and stop positions of a gene, then there is likely disagreement on the number and location of exons and introns within the gene model as well. In the case of partially supported regions, there is roughly 50% to 60% agreement on start and stop positions, but that may come as less of a surprise as there is already disgreement on presence by definition. Gene finding behaviour in regions from *T.harzianum* and



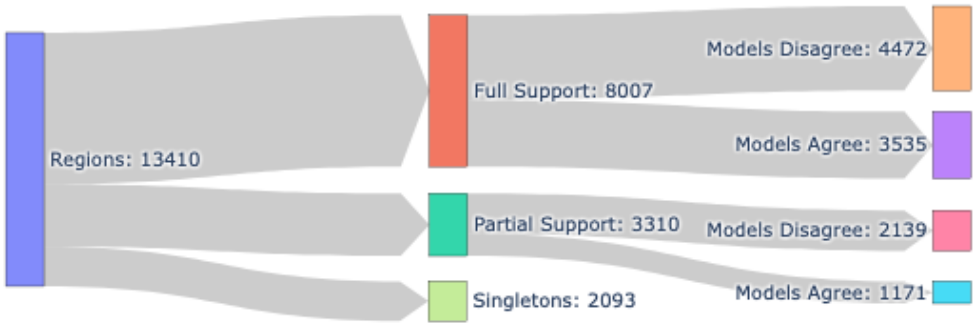
(a) DC1



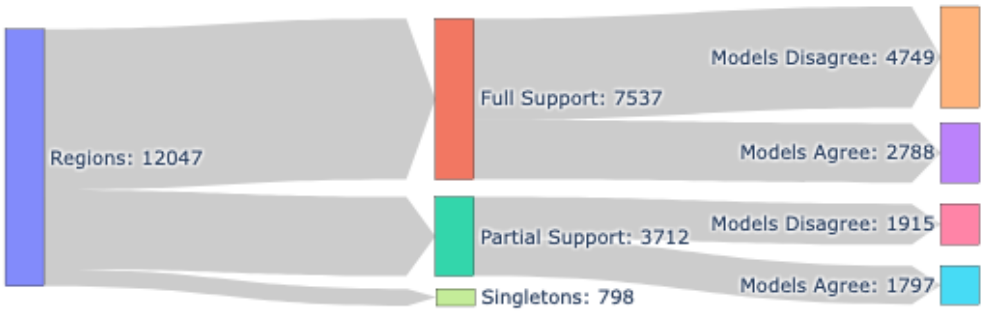
(b) Tsth20



(c) *T. reesei*



(d) *T. harzianum*



(e) *T. virens*

Figure 5.5: Figures showing breakdowns of genomic regions identified by the region finding process. Regions, in blue, are categorized based on support from gene finders. Regions with supporting predictions from all gene finders included in this analysis are labelled with 'Full Support' and colored red. Fully supported regions are then broken down into regions where gene models agree on the start and stop positions of the genes (purple), and those that do not (orange). Regions labelled with 'Partial Support' (turquoise) are regions with supporting gene predictions from two gene finders. Partially supported regions are also broken down into regions in which gene finders agree on the start and stop positions of the gene (cyan), and those that do not agree (pink). Regions with gene predictions from only one gene finder are labelled as singletons (green).

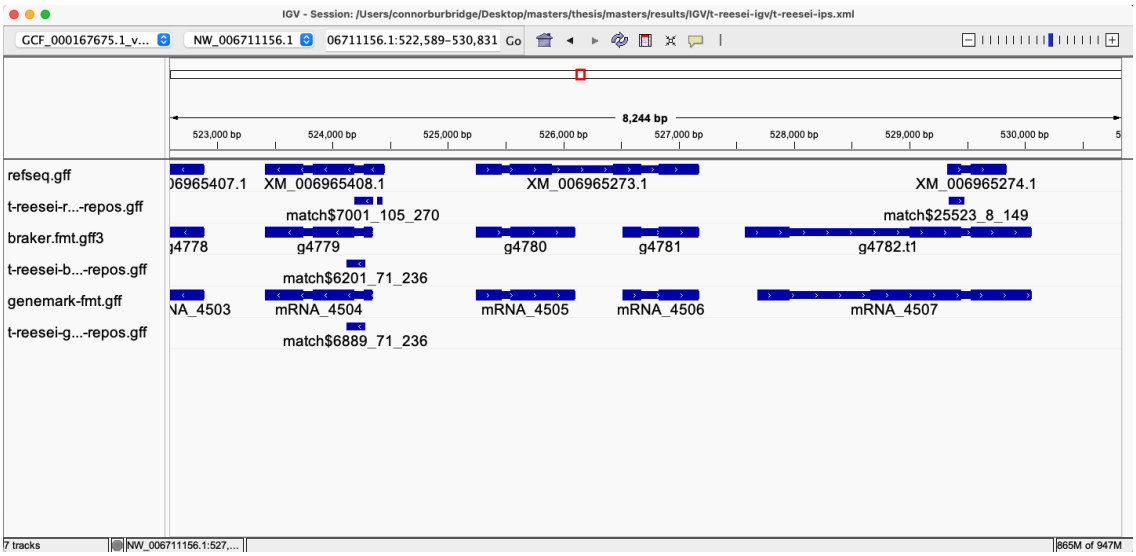


Figure 5.6: An IGV screenshot from *T. reesei* showing a region containing five gene predictions.

T. virens differ from the other assemblies in the split between fully and partially supported regions. There seems to be fewer regions with full support from all gene finders and the reason why is unclear. The gene models in each region, whether fully or partially supported, still tend to disagree on start and stop positions of the genes more often than agree.

It is also worthwhile investigating some potentially interesting cases of strange gene calls in regions. One such case is when a region contains more than three individual predicted genes. The null hypothesis, so to speak, is that if all gene finders agree exactly on the presence and position of a gene in a region, then one should expect exactly three predictions in that region. We observe that this is not always true. Cases of more than three gene predictions in a region were present in regions from all processed assemblies. Figure 5.6 shows an example of one of these regions, in which the single gene predicted by RefSeq spans multiple gene predictions from both Braker2 and GeneMark. In the case of DC1 and Tsth20, there were very few regions that matched this scenario, with 19 and 6 regions containing more than 3 gene predictions, respectively. Conversly, *T. reesei, T.harzianum and T.virens* contain many such regions, with assemblies reporting 546, 899 and 521 regions with greater than three gene predictions, respectively. While having predictions from each gene finding tool present in a region is a strong indicator for the presence of a gene, having more than one gene prediction per gene finder raises questions about which model (or models) is correct and why disagreement exists.

Another interesting set of criteria to investigate is whether or not gene finders always predict genes on the same strand within a region. We observe that regions containing predictions on different strands do exist, and are observed in predictions from all assemblies included in this analysis. Figure 5.7 shows an example of a region containing gene predictions on opposing strands. As in the case of regions with more than three gene predictions, DC1 and Tsth20 have fewer mixed strand regions with 29 regions identified in DC1 and 46

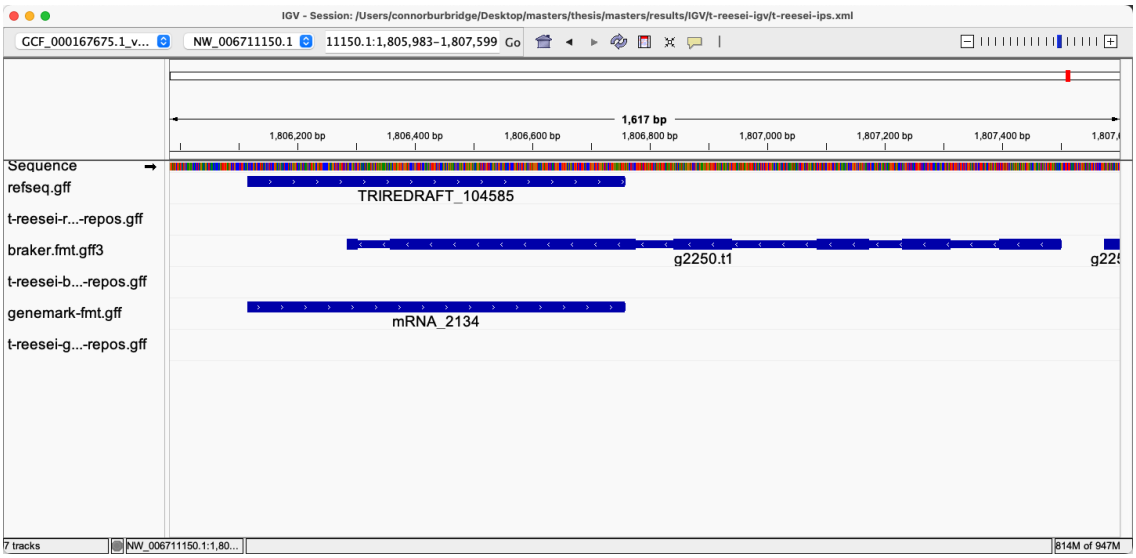


Figure 5.7: An IGV screenshot of *T. reesei* showing a region which contains gene predictions on opposite strands.

regions identified in Tsht20. Results from *T. reesei, T. harzianum* and *T. virens* show more regions in which this property is true, with region counts of 203, 533 and 293 respectively. Under the assumption that gene finders should predict the same genes on the same strands, it is unexpected to find so many of these cases, although it is possible that these overlapping sense and anti-sense genes exist naturally[40].

In summary, gene finders agree partially or completely on the presence of a gene in the vast majority of cases. While gene finders generally agree on the presence of a gene, they tend to disagree on the underlying gene model more often than they agree, except in in the case of DC1 and Tsth20. This is likely due to only including two gene finders in their analysis rather than three, resulting in fewer opportunities for disagreement. This observation is true when applied to the start and stop positions of the gene, but further investigation and comparison of intronic and exonic sequences between genes may provide more insight. It is also possible that genome quality may affect agreement between gene finders as the underlying model may be trained on contigs that are larger or of different structure than the assembly it is applied to. This is particularly true in the cases of DC1 and Tsth20, which are composed of larger contigs in comparison to the assemblies from NCBI. Finally, regions identified in this analysis do not always fit the ideal scenario of one gene prediction from each tool per region. Regions in which there are more than three gene predictions were observed in all assemblies included in this work. In addition, we observe cases where gene finders predict genes on different strands.

5.8 Genes in Regions of Anomalous GC Content

Table 5.8 shows the results of applying the same region finding process from earlier to segments of the genome identified as AT-rich. AT-rich is defined as a region of genomic sequence with percent GC composition

Assembly	Full Support	Partial Support	Singletons	No. Genes
DC1	11	N/A	20	42
Tsth20	2	N/A	9	13
T. reesei	25	18	54	194
T. harzianum	26	43	68	265
T. virens	8	11	0	49

Table 5.8: Total number of regions identified in each assembly followed by counts of regions (both agreement and singletons) in regions of AT-rich genomic sequence. It is important to note that in the cases of DC1 and Tsth20, full support indicates supporting gene predictions from both GeneMark and Braker2.

less than 28% as determined in Section 5.1. In comparison to results from Section 5.7, we see that overall there are far fewer regions with gene predictions in AT-rich genomic segements. In DC1, Tsth20, there are very few regions with full support from both Braker2 and GeneMark, but more singletons. Again, as in Section 5.7, there are no regions with partial support as only two gene finding tools were applied to those assemblies. *T. reesei* and *T. harzianum* report more regions in AT-rich genomic sequence than the other assemblies, but with the majoirty of regions belonging to the singleton category. *T. virens* is an interesting case, reporting a similar numbers of regions and genes in AT-rich genomic sequence as DC1 and Tsth20. *T. virens* is also the only assembly to report zero singleton gene predictions. Why *T. virens* differs from the other RefSeq assemblies is unclear. In general, there are few regions with gene predictions in AT-rich genomic segments, and within these regions, the majority of cases are isolated singleton gene predictions. These observations differ greatly from those made in nucleotide composition agnostic approach in section 5.7.

In addition to a breakdown of regions in AT-rich genomic sequence, understanding which gene finders predict more or fewer genes in these regions may be of interest. It is possible that an HMM, trained on genomic sequence with varying nucleotide content, may predict genes differently to an HMM trained only on sequences with uniformally distributed nucleotide composition as the variation in nucleotide composition may affect the various states and relationships between them in an HMM. Table 5.9 shows the number of genes predicted by each gene finding tool in regions of AT-rich genomic sequence. GeneMark appears to predict the fewest genes in AT-rich regions, while RefSeq appears to predict the most. Braker2 lies somewhere in the middle. Again, *T. virens* appears as an odd case, with very few predictions from all gene finders. Why *T. virens* differs from the other RefSeq assemblies is unclear.

Finally, to test the probability of any given gene prediction falling in an AT-rich genomic sequence, a two-sided binomial test was performed to determine if the number of genes predicted in AT-rich sequences is proportional to fraction of genomic sequence they comprise. The null hypothesis in this case is that the number of genes in AT-rich sequence is proportional to the total AT-rich sequence in the genome. The results of the test are shown in Table 5.10. In all cases, it appears that the gene finding tools selected for this analysis

Assembly	Braker2	GeneMark	RefSeq
DC1	31	11	N/A
Tsth20	11	2	N/A
T. reesei	39	48	107
T. harzianum	81	30	154
T.virens	21	8	20

Table 5.9: Number of genes predicted by Braker2, GeneMark and RefSeq in AT-rich genomic sequence from each assembly. It is important to note that in the cases of DC1 and Tsth20, full support indicates supporting gene predictions from both GeneMark and Braker2.

Tool	DC1	Tsth20	*T. reesei*	*T. harzianum*	*T. virens*
Braker2	$9.56 * 10^{-181}$	$1.14 * 10^{-259}$	$2.68 * 10^{-96}$	$4.05 * 10^{-140}$	$1.35 * 10^{-35}$
GeneMark	$5.12 * 10^{-216}$	0.0	$5.66 * 10^{-49}$	$5.37 * 10^{-219}$	$5.31 * 10^{-35}$
RefSeq	N/A	N/A	$1.29 * 10^{-49}$	$2.44 * 10^{-205}$	$7.40 * 10^{-33}$

Table 5.10: p-values produced from a two-sided binomial test for each combination of tool and assembly.

do not predict the same proportion of genes in AT-rich genomic sequence as in typical genomic sequence.

In summary, very few regions with gene predictions are present in AT-rich genomic sequence. Additionally, genes predicted in these regions tend to be isolated and not supported by other gene finders, although some agreement is observed. In terms of number of genes predicted, RefSeq tends to predict the most genes in these AT-rich regions while GeneMark predicts the fewest. Lastly, the selected gene finding tools do not predict genes in AT-rich sequences in proportion to the fraction of genomic sequence they comprise.

5.9 InterProScan as Supporting Evidence for Predicted Genes

Pfam hits from InterProScan analysis are presented in Table 5.11, from which we can identify one major trend. In general, roughly 73-76% of proteins predicted by Braker2, GeneMark and RefSeq contain a match to a Pfam entry, except in the case of *T. harzianum*, which reports a considerably lower proportion of genes with Pfam matches in the RefSeq dataset. Why the proportion of RefSeq proteins with Pfam matches in *T. harzianum* is so low is unknown. This is promising performance for the gene finders as the RefSeq annotations demonstrate similar proportions of Pfam hits to predicted proteins. The total counts may be decieving however, as predictions from Braker2 may result in more than one protein product per gene, whereas in the case of GeneMark, only one protein is produced per gene model. From visual inspection of Pfam hits mapped back to the references, it appears that in general, when gene finders agree that a gene

is present, InterProScan reports the same Pfam match in all three predictions. An example of agreement between Braker2, GeneMark, RefSeq and InterProScan is shown in Figure 5.8. It is important to note that the position of the Pfam match in IGV does not indicate the true position of the Pfam match in the gene, but provides an indication of the presence of Pfam matches. Pfam matches are offset from the start of the gene based on the start and end position of the Pfam match in the protein sequence. For example, if a Pfam match has a start position 10 amino acids into the protein sequence, the corresponding start position in the resulting GFF is 10bp downstream from the start of the gene.

Assembly	Braker2	GeneMark	RefSeq
DC1	$\frac{10676}{14479}$	$\frac{8416}{11354}$	N/A
Tsth20	$\frac{11389}{15546}$	$\frac{9168}{12373}$	N/A
T. reesei	$\frac{8471}{11704}$	$\frac{6990}{9196}$	$\frac{6964}{9111}$
T. harzianum	$\frac{11370}{15408}$	$\frac{9061}{12164}$	$\frac{9293}{14065}$
T. virens	$\frac{11249}{15062}$	$\frac{8871}{11866}$	$\frac{9062}{12383}$

Table 5.11: Table showing the fractions of predicted genes with Pfam annotations from InterProScan as a percent

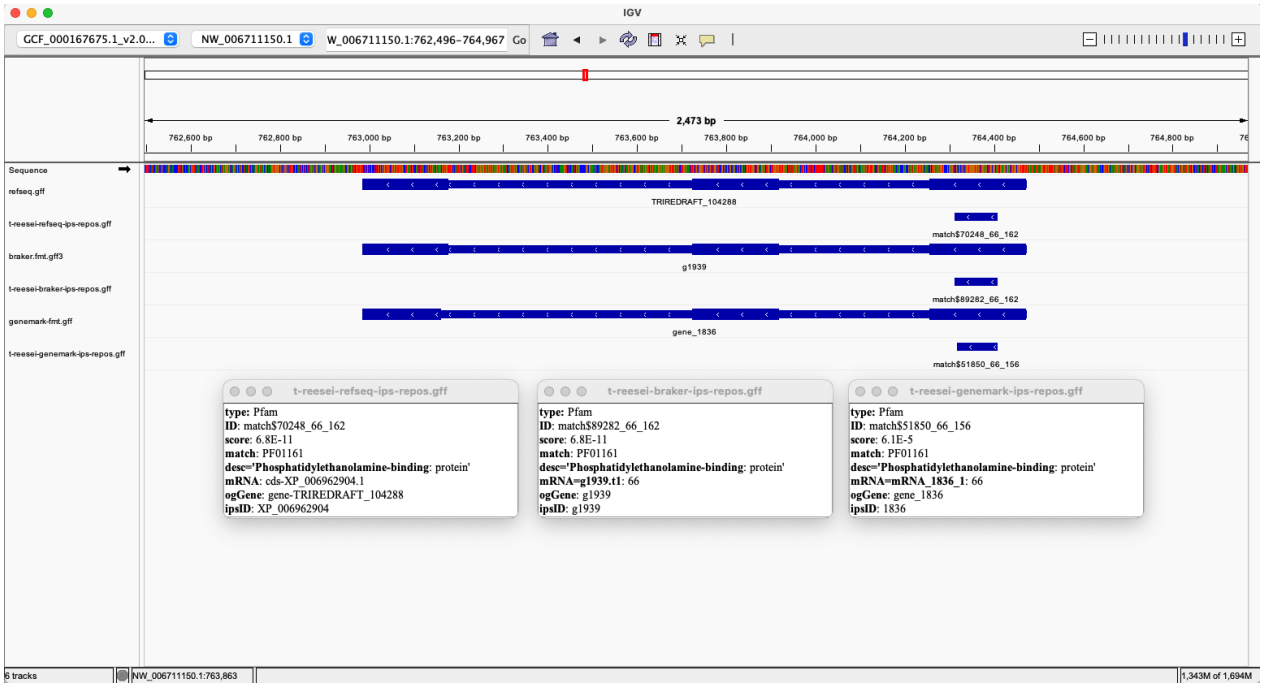


Figure 5.8: An IGV capture showing complete agreement between gene finders for both gene model and protein Pfam hits. The longer segments are predicted genes and the smaller segments are annotated Pfam matches, all with agreeing start and stop positions. The **desc** in each sub-window indicate agreeing Pfam annotations.

While cases of complete agreement are abundant, cases of disagreemnt also exist and in strange forms. In many cases, while the gene finders agree on the presence of a gene, only the RefSeq protein product contains

a match to the Pfam database. Why this may be the case is unclear, and may warrant further investigation. There are also many cases in which InterProScan reports the same Pfam hits for individual proteins, but the gene models in the region do not agree. There are even cases such as the region shown in Figure 5.9, where Braker2 and GeneMark agree that two genes and their associated proteins and Pfam matches are separate, but RefSeq only reports one gene with multiple Pfam hits. There are also several cases where two tools are in agreement with proteins containing Pfam hits while another is not. Even more interesting are cases such as the one shown in Figure 5.10, in which Braker and GeneMark predictions contain Pfam matches while RefSeq does not report a gene at all. This may be the due to experimental data used in the RefSeq training process or a result of curation. Regardless of the tool, these cases demonstrate well that gene finders are not always in agreement even on well known proteins, to the point that predictions are not present even though protein products from other gene finders contain known Pfam matches.

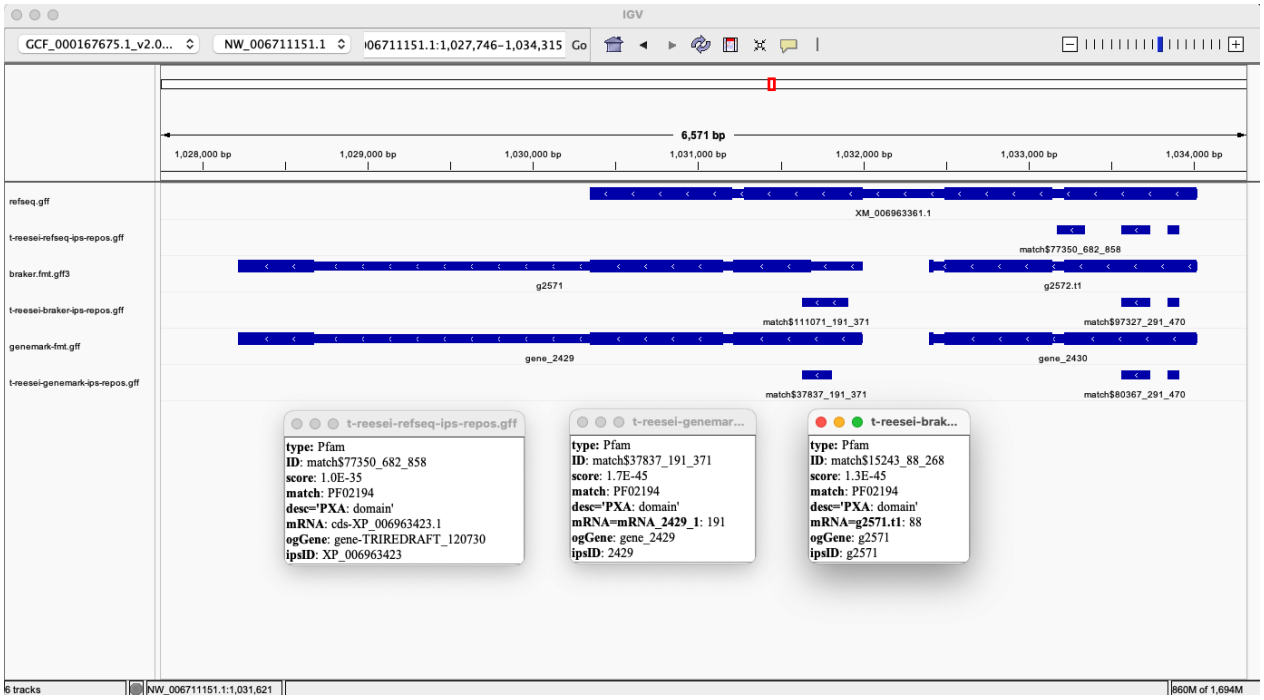


Figure 5.9: An IGV capture showing Braker2 and GeneMark reporting two genes and their resulting proteins and Pfam hits as separate, while RefSeq reports one gene, one protein and three Pfam matches.

In summary, the protein products from Braker2 and GeneMark predictions do contain matches to the Pfam database. The proportions of matches to total proteins are similar to that of the RefSeq annotation, sitting between 65 and 75 percent. While Pfam hits to Braker2, GeneMark and RefSeq proteins generally agree, we do observe regions in which there is disagreement in serveral forms.

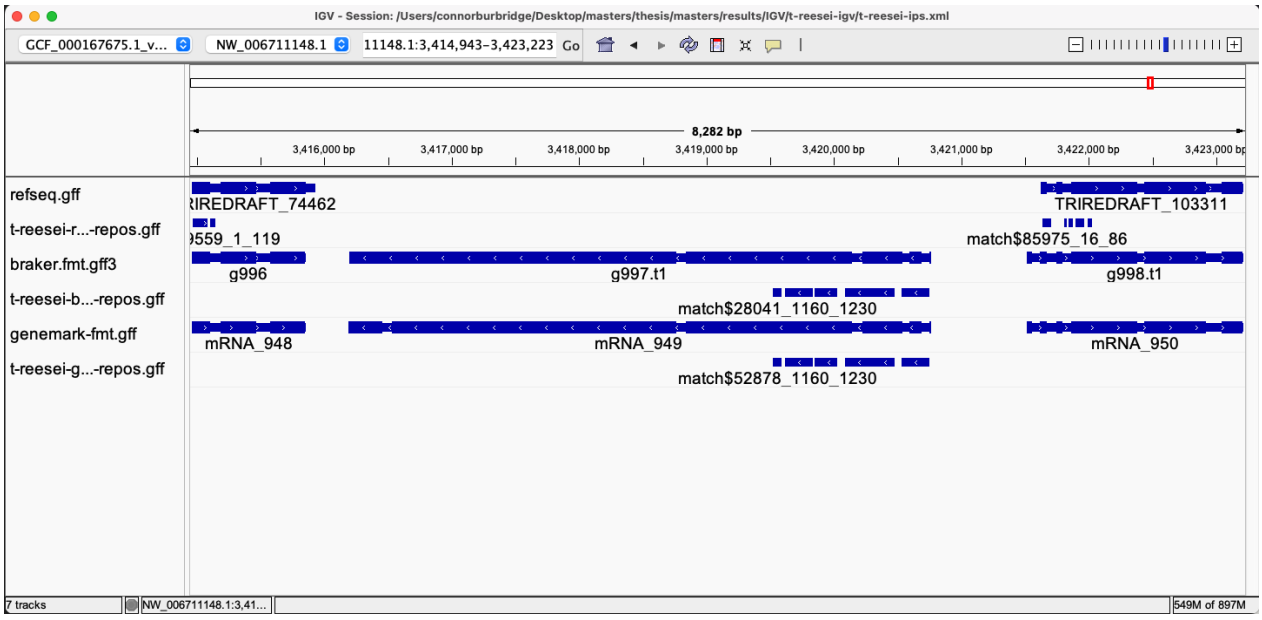


Figure 5.10: An IGV capture showing a scenario where GeneMark and Braker2 agree on a gene model with supporting Pfam evidence and RefSeq does not report any gene.

5.10 Selection of Gene Finding Tool

With all of these results, it makes sense to explore the question of which gene finding tool one should choose for optimal gene prediction performance. Comparisons drawn in this section are made in the context of *Trichoderma* assemblies and may not extend to other datasets. Results from this work are summarized in table 5.12. Ignoring availability and use of the gene finding tools, it would appear that RefSeq performs the best in the remaining categories, earning top marks in every category except in it's ability to predict very short genes. Braker2 earns second place; however, this does not capture Braker2's failure in predicting accurate numbers of genes in DC1, Tsth20, *T. harzianum* and *T. virens*. GeneMark comes in last, excelling only in number of genes predicted and Pfam support for the genes that it predicts.

Relating these observations to use-case scenarios, in the case that your organism of interest has a RefSeq annotation associated with it, the RefSeq gene prediction process appears to produce the best set of predictions. If users also have experimental evidence, such as RNAseq data under experimental conditions, it may be worthwhile training a Braker2 model and predicting genes with Braker2 to supplement the already well performing RefSeq gene predictions. If the organism of interest is not a RefSeq individual but training data is available for that organism, Braker2 is the next best option, although it is important to note that the application of a trained Braker2 prediction model to an organism from which the training data did not originate is not advised based on the results presented in this work (see 5.3). While it is true that the *T. reesei* genome differs from other *Trichoderma* genomes, it's status as a representative RefSeq organism makes it somewhat of a gold standard. In this case, applying a gene model trained using evidence from the gold

standard produces biased numbers of genes predicted in other *Trichoderma* genomes. While Braker2 technically scores the second highest, users must be very careful when selecting training data, and ensure that the training data either comes from the organism of interest, or comes from a very closely related organism with a highly similar genome. In the case that no appropriate training data is available, GeneMark is still an option, and users can be confident that the tool predicts a reasonably accurate number of genes with supporting Pfam matches. It is also important to note that GeneMark does not perform as well in AT-rich regions as Braker2 and RefSeq, does not predict isoforms, and systematically fails to predict some BUSCO orthologs.

When availability of a gene finding tool becomes a concern and RefSeq is not considered, the scores drop significantly for Braker2 and GeneMark as seen in the final row of table 5.12. In this situation, Braker2 still outperforms GeneMark. If the organism of interest is not considered a representative RefSeq individual, but supporting evidence specific to that organism or a very closely related organism is available, a trained Braker2 prediction model will perform well. Again, in the case that the organism is not a RefSeq individual and no appropriate training data is available, GeneMark is still a reasonable option even with the previously identified caveats.

In summary, these results indicate that if your organism is a RefSeq organism, use the RefSeq annotation. If no RefSeq predictions are available but appropriate training data is, one should use Braker2. If the training data is of questionable similarity or not available at all, users can fall back on *ab initio* gene finders such as GeneMark, which while not ideal, still predict genes with supporting evidence.

Category	Braker2	GeneMark	RefSeq
Availability	3	3	0
Ease of install	1	2	0
Ease of use	3	3	0
# of genes predicted	0	3	3
# of transcripts predicted	3	0	2
Predicts shortest genes	2	1	0
Predicts more shorter genes	1	0	3
BUSCO Performance	2	1	3
Performance in AT-rich sequence	2	1	3
Predictions with InterProScan support	3	3	3
Final Score (Publicly Available)	20	17	N/A
Final Score (Ignoring Availability)	13	9	17

Table 5.12: Table with scores attirbuted to performance of each gene finder in several categories. The score definitions for performance are as follows: 0 - fail, 1 - pass, 2 - good, 3-excellent. Since RefSeq is not publicly available, it is marked as N/A in the publicly available final scores. The dashed line separates categories associated with availability and use from categories describing gene prediction performance.

5.11 Conclusions

References

[1] *Kolmogorov–Smirnov Test*, pages 283–287. Springer New York, New York, NY, 2008.

[2] Jill Adams. Complex Genomes: Shotgun Sequencing. *Nature Education*, 1(1):186, 2008.

[3] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, oct 1990.

[4] Derek W Barnett, Erik K Garrison, Aaron R Quinlan, Michael P Strömberg, and Gabor T Marth. BamTools: a C++ API and toolkit for analyzing and managing BAM files. *Bioinformatics (Oxford, England)*, 27(12):1691–1692, jun 2011.

[5] Simon Bennett. Solexa Ltd. *Pharmacogenomics*, 5(4):433–438, jun 2004.

[6] Anthony M Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, aug 2014.

[7] Mark Borodovsky and Alex Lomsadze. Eukaryotic gene prediction using GeneMark.hmm-E and GeneMark-ES. *Current Protocols in Bioinformatics*, Chapter 4(SUPPL.35):4.6.1–4.6.10, sep 2011.

[8] Tomáš Brůna, Katharina J Hoff, Alexandre Lomsadze, Mario Stanke, and Mark Borodovsky. BRAKER2: automatic eukaryotic genome annotation with GeneMark-EP+ and AUGUSTUS supported by a protein database. *NAR genomics and bioinformatics*, 3(1):lqaa108, mar 2021.

[9] Brad Chapman. BCBio Python Package.

[10] Phillip E C Compeau, Pavel A Pevzner, and Glenn Tesler. How to apply de Bruijn graphs to genome assembly. *Nature biotechnology*, 29(11):987–991, nov 2011.

[11] Rina Dechter. chapter 12 - Temporal Constraint Networks. In Rina Dechter, editor, *Constraint Processing*, The Morgan Kaufmann Series in Artificial Intelligence, pages 333–362. Morgan Kaufmann, San Francisco, 2003.

[12] Girum Fitihamlak Ejigu and Jaehee Jung. Review on the Computational Genome Annotation of Sequences Obtained by Next-Generation Sequencing. *Biology*, 9(9), sep 2020.

[13] Python Software Foundation. Python.

[14] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351, 2016.

[15] Alexey Gurevich, Vladislav Saveliev, Nikolay Vyahhi, and Glenn Tesler. QUAST: quality assessment tool for genome assemblies. *Bioinformatics (Oxford, England)*, 29(8):1072–1075, apr 2013.

[16] Jiang Hu, Junpeng Fan, Zongyi Sun, and Shanlin Liu. NextPolish: a fast and efficient genome polishing tool for long-read assembly. *Bioinformatics*, 36(7):2253–2255, apr 2020.

[17] Jiang Hu, Zhuo Wang, Zongyi Sun, Benxia Hu, Adeola Oluwakemi Ayoola, Fan Liang, Jingjing Li, José R Sandoval, David N Cooper, Kai Ye, Jue Ruan, Chuan-Le Xiao, Depeng Wang, Dong-Dong Wu, and Sheng Wang. NextDenovo: an efficient error correction and accurate assembly tool for noisy long reads. *Genome Biology*, 25(1):107, 2024.

[18] Philip Jones, David Binns, Hsin-Yu Chang, Matthew Fraser, Weizhong Li, Craig McAnulla, Hamish McWilliam, John Maslen, Alex Mitchell, Gift Nuka, Sebastien Pesseat, Antony F Quinn, Amaia Sangrador-Vegas, Maxim Scheremetjew, Siew-Yit Yong, Rodrigo Lopez, and Sarah Hunter. InterProScan 5: genome-scale protein function classification. *Bioinformatics*, 30(9):1236–1240, 2014.

[19] Christian P Kubicek, Andrei S Steindorff, Komal Chenthamara, Gelsomina Manganiello, Bernard Henrissat, Jian Zhang, Feng Cai, Alexey G Kopchinskiy, Eva M Kubicek, Alan Kuo, Riccardo Baroncelli, Sabrina Sarrocco, Eliane Ferreira Noronha, Giovanni Vannacci, Qirong Shen, Igor V Grigoriev, and Irina S Druzhinina. Evolution and comparative genomics of the most common Trichoderma species. *BMC Genomics*, 20(1):485, 2019.

[20] Alexandre Lomsadze, Vardges Ter-Hovhannisyan, Yury O Chernoff, and Mark Borodovsky. Gene identification in novel eukaryotic genomes by self-training algorithm. *Nucleic Acids Research*, 33(20):6494–6506, 2005.

[21] Vivien Marx. Method of the year: long-read sequencing. *Nature Methods*, 20(1):6–11, 2023.

[22] Niranjan Nagarajan and Mihai Pop. Sequence assembly demystified. *Nature Reviews Genetics*, 14(3):157–167, 2013.

[23] NCBI. NCBI Eukaryotic Annotation Pipeline, 2024.

[24] NCBI. NCBI SRA Toolkit, 2025.

[25] Geo Pertea and Mihaela Pertea. GFF Utilities: GffRead and GffCompare. *F1000Research*, 9, 2020.

[26] Mihaela Pertea, Geo M Pertea, Corina M Antonescu, Tsung-Cheng Chang, Joshua T Mendell, and Steven L Salzberg. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature Biotechnology*, 33(3):290–295, 2015.

[27] Miroslav Plohl, Nevenka Meštrović, and Brankica Mravinac. Centromere identity from the DNA point of view. *Chromosoma*, 123(4):313–325, aug 2014.

[28] Aaron R Quinlan and Ira M Hall. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010.

[29] P Rice, I Longden, and A Bleasby. EMBOSS: the European Molecular Biology Open Software Suite. *Trends in genetics : TIG*, 16(6):276–277, jun 2000.

[30] James T Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz, and Jill P Mesirov. Integrative genomics viewer. *Nature Biotechnology*, 29(1):24–26, 2011.

[31] Biancamaria Senizza, Fabrizio Araniti, Simon Lewin, Sonja Wende, Steffen Kolb, and Luigi Lucini. Trichoderma spp.-mediated mitigation of heat, drought, and their combination on the Arabidopsis thaliana holobiont: a metabolomics and metabarcoding approach. *Frontiers in Plant Science*, 14, 2023.

[32] Felipe A Simão, Robert M Waterhouse, Panagiotis Ioannidis, Evgenia V Kriventseva, and Evgeny M Zdobnov. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*, 31(19):3210–3212, 2015.

[33] R Keith Slotkin. The case for not masking away repetitive DNA. *Mobile DNA*, 9(1):15, 2018.

[34] Mario Stanke, Ana Tzvetkova, and Burkhard Morgenstern. AUGUSTUS at EGASP: using EST, protein and genomic alignments for improved gene prediction in the human genome. *Genome Biology*, 7(1):S11, 2006.

[35] Ioannis S Arvanitoyannis Theodoros H. Varzakas and Haralambos Baltas. The Politics and Science Behind GMO Acceptance. *Critical Reviews in Food Science and Nutrition*, 47(4):335–361, 2007.

[36] Todd J Treangen and Steven L Salzberg. Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nature reviews. Genetics*, 13(1):36–46, nov 2011.

[37] Yunhao Wang, Yue Zhao, Audrey Bollas, Yuru Wang, and Kin Fai Au. Nanopore sequencing technology, bioinformatics and applications. *Nature Biotechnology*, 39(11):1348–1365, 2021.

[38] David J Winter, Austen R D Ganley, Carolyn A Young, Ivan Liachko, Christopher L Schardl, Pierre-Yves Dupont, Daniel Berry, Arvina Ram, Barry Scott, and Murray P Cox. Repeat elements organise 3D genome structure and mediate transcription in the filamentous fungus Epichloë festucae. *PLOS Genetics*, 14(10):1–29, 2018.

[39] Sheridan L Woo, Rosa Hermosa, Matteo Lorito, and Enrique Monte. Trichoderma: a multipurpose, plant-beneficial microorganism for eco-sustainable agriculture. *Nature Reviews Microbiology*, 21(5):312–326, 2023.

[40] Bradley W Wright, Mark P Molloy, and Paul R Jaschke. Overlapping genes in natural and engineered genomes. *Nature Reviews Genetics*, 23(3):154–168, 2022.