4 Data and Methodology

4.1 Methodology Overview

The general processing steps performed in this work can be grouped into four processing stages, or workflows. A simplified overview of these stages and the flow of data between them is shown in Figure 4.1. In ascending order, each processing stage is discussed in more detail in Sections 4.2, 4.4, 4.9, and .

4.2 Sequence Processing and Assembly Workflow

The sequence processing and assembly workflow is shown in Figure 4.2. Processing steps are represented by boxes, results of interest generated by this work are represented by folders, and datasets that were not produced by this work are represented by cylinders. Other workflows that use datasets produced by this workflow are depicted as arrows outside of the brown sub-graph.

Illumina©™[5] sequences were first trimmed and filtered using Trimmomatic[6] to remove adapter content and low-quality sequences, as low-quality sequences may affect the assembly process. The processed Illumina©™sequences and Nanopore©™[39] sequences were then supplied to the hybrid genome assembly tool NextDenovo[17]. The processed Illumina©™data was then re-used to polish the assemblies with NextPolish[16]. The resulting assemblies were then analyzed to identify AT-rich genomic sequence, analyzed with QUAST for general assembly metrics, and then used as input in the gene prediction workflow.

The processing steps Trimmomatic filtering, NextDenovo assembly, NextPolish Polishing and QUAST assembly assessment are described in Section 4.3. The AT-rich sequence identification process is described later in Section 4.15.

4.3 Sequence Pre-processing and Assembly

Genomic sequences used for assembly were generated for DC1 and Tsth20 using Nanopore©™[39] and Illumina©™[5] sequencing technologies by Brendan Ashby at the Global Institute for Food Security at the University of Saskatchewan. 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]

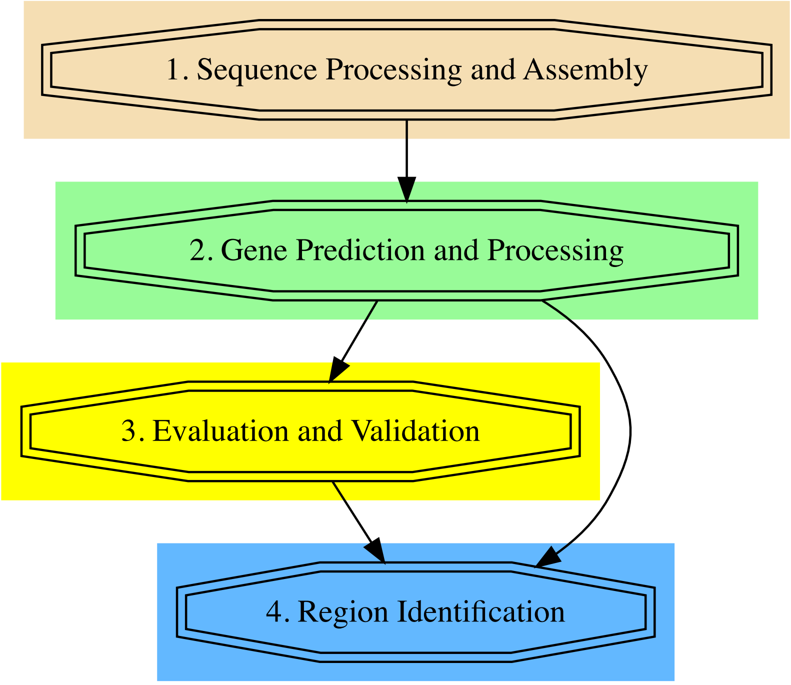
1. Sequence Processing and Assembly
2. Gene Prediction and Processing
3. Evaluation and Validation
4. Region Identification

Figure 4.1: A simplified workflow used in this work. Each processing stage is is represented by an octagon, and the flow of data are indicated by directed edges between them. In this figure and subsequent figures, the sequence processing stage is coloured brown, the gene prediction and processing stage is coloured green, the evaluation and validation stage is coloured yellow, and the region identification process is coloured blue.

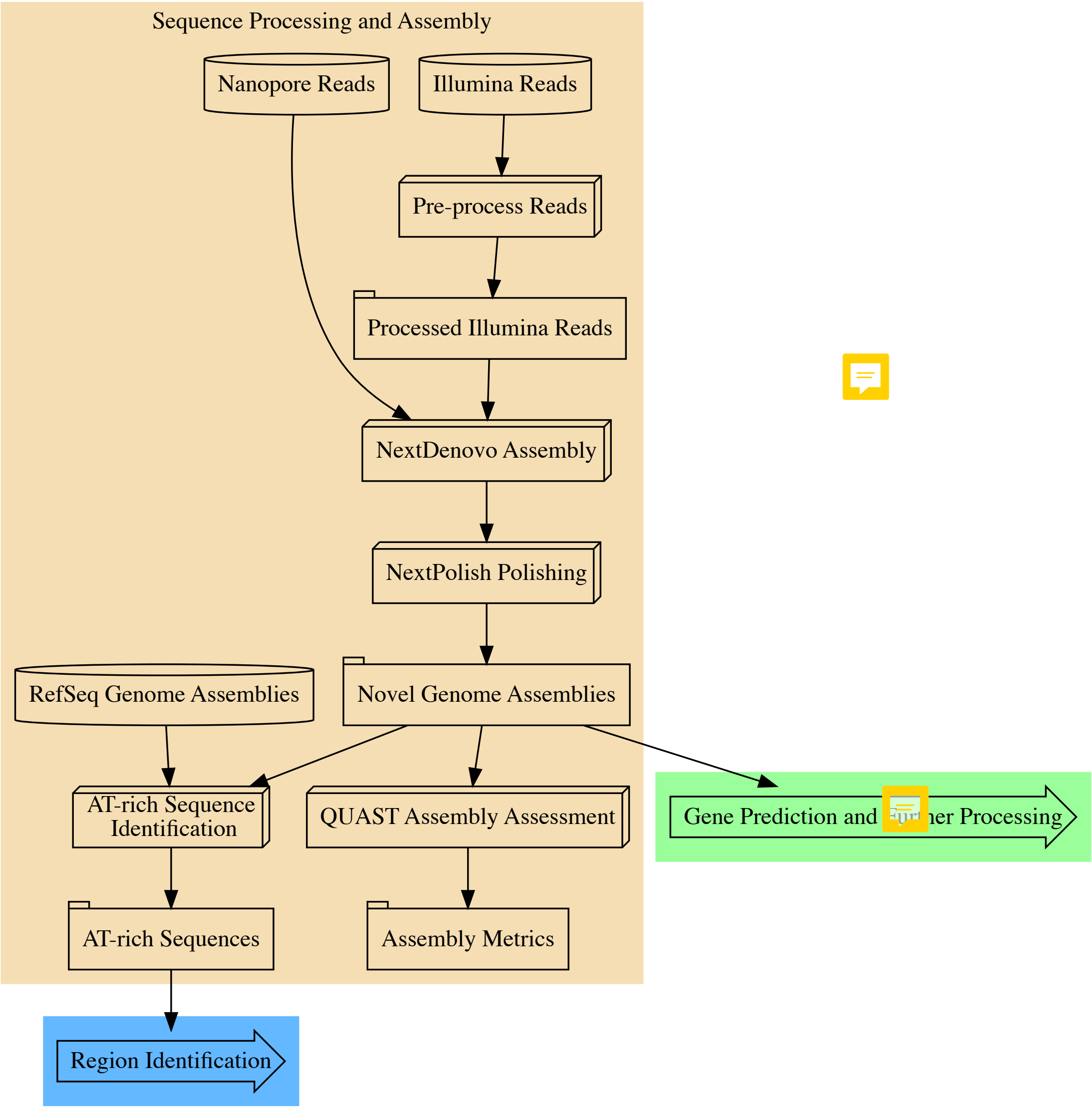

Figure 4.2: A workflow (in brown) depicting the steps taken to process input sequences, generate assemblies, and post-process those assemblies for use in other workflows. Processing steps are represented by boxes, datasets of interest that were produced by this work are represented by folders, and datasets that were not produced in this work are represented by cylinders. Other workflows are represented by arrows outside of the brown sub-graph. Directed edges indicate the flow of data.

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.

4.4 Overview of Gene Prediction and Further Processing

To better understand the methods used in the gene prediction and ~~CDS analysis~~ workflow, the steps have been grouped together in Figure 4.3. Processing steps are represented by boxes, results of interest generated by this work are represented by folders, and datasets that were not produced in this work are represented by cylinders. Other workflows that use datasets produced by this workflow are depicted as arrows outside of the green sub-graph.

First, RefSeq assemblies and their associated annotations of closely related *Trichoderma* species were selected from NCBI for training and comparison. Both ~~and~~ *ab initio* (GeneMark[7]) and evidence-based (Braker2[8]) gene finders were selected for comparison to evaluate gene prediction behaviour of both methods in the context of *Trichoderma* genomes. Gene prediction was performed using both gene finders on the RefSeq genomes and the genomes of DC1 and Tsth20. Coding sequences (CDS) output by the gene finders were then examined to better understand distributions of CDS lengths produced by different gene finders. Finally, all gene predictions were then supplied to the evaluation and validation stage as well as the region identification stage.

Selection of NCBI datasets for training and comparison are further discussed in Section 4.5. The GeneMark gene prediction process is described in Section 4.6, while the Braker2 gene prediction process is described in Section 4.7. The CDS processing step is described in Section 4.8.

4.5 Selection of Datasets from NCBI

When evaluating gene-finders, it is important compare gene predictions to a standard, or control dataset, as a method of ground-truthing. Such datasets include reference genome sequences and their associated gene predictions from NCBI as used in this work. In addition to comparing against control datasets, RNAseq data can also be used as training data for various tools as well, such as the evidence-based training used by Braker2. In this work, four forms of these inputs were selected from NCBI for comparison and training. The first two datasets selected were assemblies and associated gene predictions of three RefSeq *Trichoderma* accessions, with those accessions being *T. virens*, *T. harzianum*, and *T. virens*. NCBI RefSeq accession IDs for those assemblies are GCF_000167675.1_v2.0, GCF_003025095.1_Triha_v1.0, and GCF_000170995.1_TRIVI_v2.0, respectively. *T. reesei* was selected as it is the most well-studied *Trichoderma* species, and widely considered to be the gold-standard for *Trichoderma* genomes. *T. harzianum* and *T. virens* were selected as additional datasets due to their prevalence in literature as well.

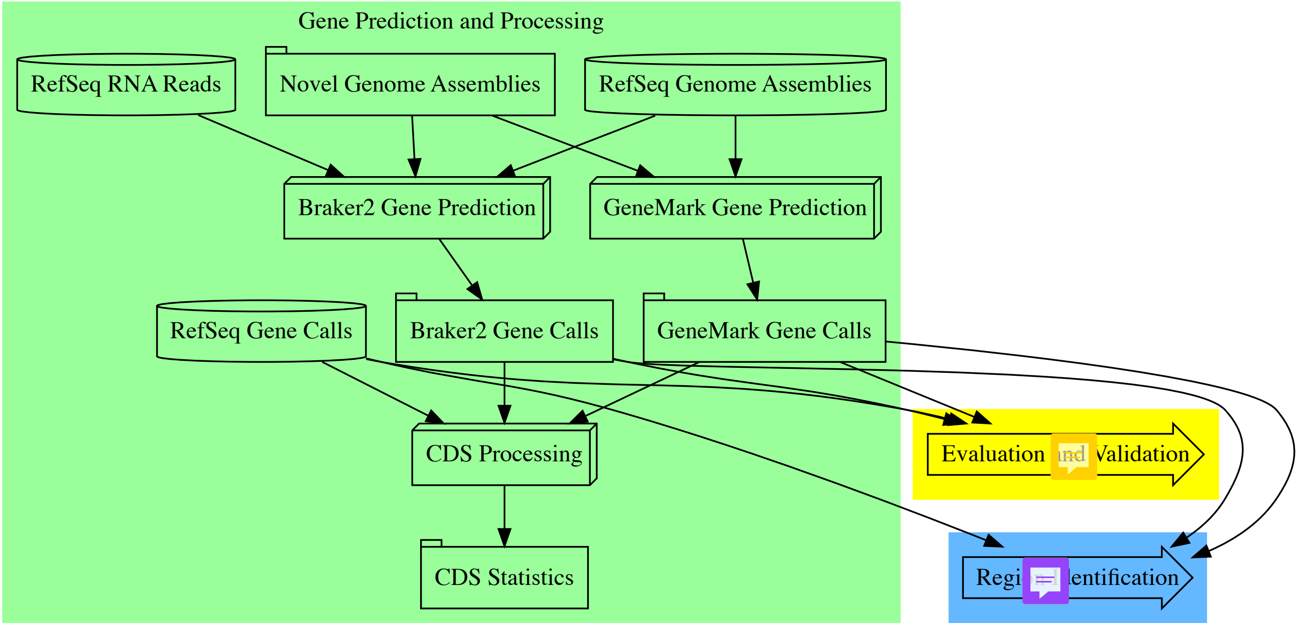

Figure 4.3: A workflow (green) depicting the steps taken to predict genes for use in other workflows and process CDS sequence lengths. Processing steps are represented by boxes, datasets of interest that were produced by this work are represented by folders, and datasets that were not produced in this work are represented by cylinders. Other workflows are represented by arrows outside of the green sub-graph. Directed edges indicate the flow of data.

The second set of selected inputs were RNAseq datasets from *T. reesei*, used as input in the training process for Braker2. The SRA accession IDs for those sequences are: SRR5229930, SRR8329344, SRR8329345, SRR8329346, and SRR8329347. Limited RNAseq datasets were used as there were few RNAseq datasets available at the beginning of this project.

The final set of inputs ~~are~~ protein sequences from RefSeq ~~individuals~~ *T. atroviride*, *Fusarium graminarium*, and *Saccharomyces cerevisiae*, which were for downstream validation of gene predictions. The NCBI accession IDs for these assemblies and associated protein sequences are GCF_000171015.1, GCF_000240135.3, and GCF_000146045.2, respectively.

4.6 Gene Prediction Using GeneMark-ES

Ab initio gene finding was performed using GeneMark-ES v4.71[7] with DC1, Tsth20 and selected RefSeq *Trichoderma* assemblies used as input. Default parameters were used in all cases except for the fungal option, which allowed the use of a GeneMark model specific to fungal genomes.

4.7 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. Illumina©™[5] RNAseq datasets from *T. reesei* were downloaded from the NCBI short-read archive using the sra-toolkit[25] 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.

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

4.8 Statistical Analysis of CDS Lengths

The distribution of gene lengths predicted by a gene-finder is an important feature to consider when selecting a gene finding tool, as genes of a particular length may be of interest to researchers, but more importantly, gene-finders may be biased to certain distributions based on their underlying models. This is important as the distribution of predicted gene lengths may also play a role in *Trichoderma's* role as an avirulent plant symbiont. These biased distributions may differ from the true distribution of CDS lengths leading to propagation of sub-optimal gene-finding results and results that may not reflect the true distribution of gene lengths. To determine, if differences exist between distributions of predicted gene lengths from each tool, the $log^{1}0$ lengths of coding sequences (CDS) were subject to a two-sample Kolmogorov-Smirnov test[1].

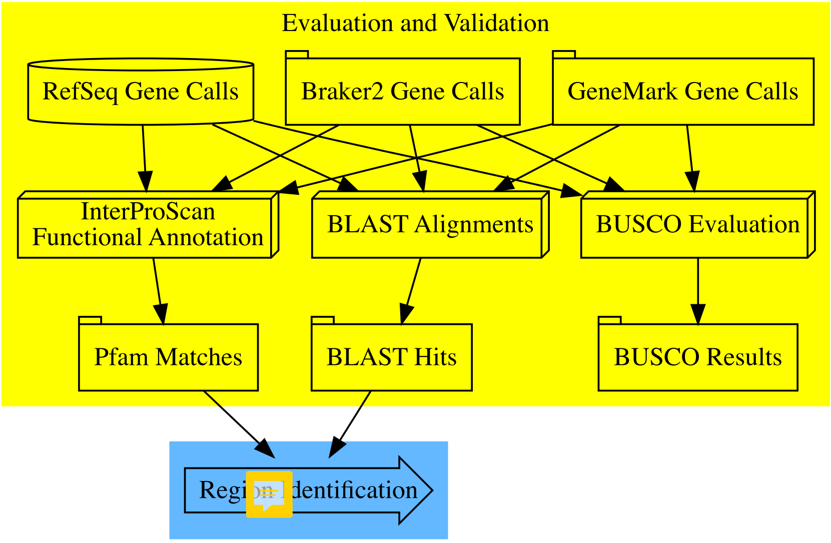

Figure 4.4: A workflow (yellow) depicting the steps taken to evaluate and validate predicted genes. Processing steps are represented by boxes, datasets of interest that were produced by this work are represented by folders, and datasets that were not produced in this work are represented by cylinders. Other workflows using results from these processing steps are represented by arrows outside of the green sub-graph. Directed edges indicate the flow of data.

4.9 Overview of Evaluation and Validation of Gene Predictions

Following the gene prediction process, prediction results can then be evaluated and ~~queried~~ against exiting datasets as a form of validation. The workflow for evaluation and validation of gene predictions is shown in Figure 4.4. Processing steps are represented by boxes, results of interest generated by this work are represented by folders, and datasets that were not produced in this work are represented by cylinders. Other workflows that use datasets produced by this workflow are depicted as arrows outside of the yellow sub-graph. The flow of data is represented by directed edges.

Coding sequences from the previous stage were evaluated and validated using several tools. Firstly, protein products from predicted CDSs were processed with InterProScan to identify Pfam domains as evidence for the validity of gene predictions. Secondly, CDSs were analyzed with BUSCO to assess prediction of conserved fungal genes. Finally, RefSeq proteins from related fungal species were used in tblastn searches of the five *Trichoderma* assemblies selected for the main body of this work.

BUSCO evaluation, InterProScan Annotation, and BLAST alignments processes are described in more detail in Sections 4.10, 4.11, and 4.12, respectively. Results from these processes are also used as inputs in the region identification workflow later.

4.10 Evaluating Gene-Finder Performance Using BUSCO

While novel gene predictions are of great interest in new subjects, it is also important to confirm that gene-finders are predicting genes that are expected to be present in the organism of interest[21]. To determine if this is true, BUSCO v5.7.1[34] 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.

4.11 Functional Annotation Using InterProScan

The presence of a functional component in a predicted gene's protein product is good evidence in favour of a gene-finder's performance. To identify functional components, the protein products from each gene finder's predictions 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 parameters. The resulting protein annotations were mapped back to their genome assemblies and processed using the region identification algorithm in section 4.14.

4.12 Ground-truthing with BLAST

The *Trichoderma* assemblies chosen for comparison were used as the subject sequences in tblastn searches with the query proteins discussed in ~~paragraph three of~~ Section 4.5. Default parameters were used in the tblastn searches. Hits from the tblastn alignments were then filtered for alignments with a minimum of 30% query coverage and 30% identity. These criteria were selected as they filter out spurious alignments while also retaining short alignments that capture functional or conserved protein coding sequence. Remaining alignments were then processed using the region identification algorithm in 4.14

4.13 Region Identification Overview

A major portion of this work was the process of identifying regions of overlapping features as a method of comparing gene calls from different assemblers. The definition of a region and the algorithm used to identify them is described in detail in Section 4.14, and an overview of the region identification workflow is shown in Figure 4.5. The top of the graph shows the gene calls produced by the workflow described in Section 4.4, which are used as the main input to the region identification process ~~along with other~~ previously

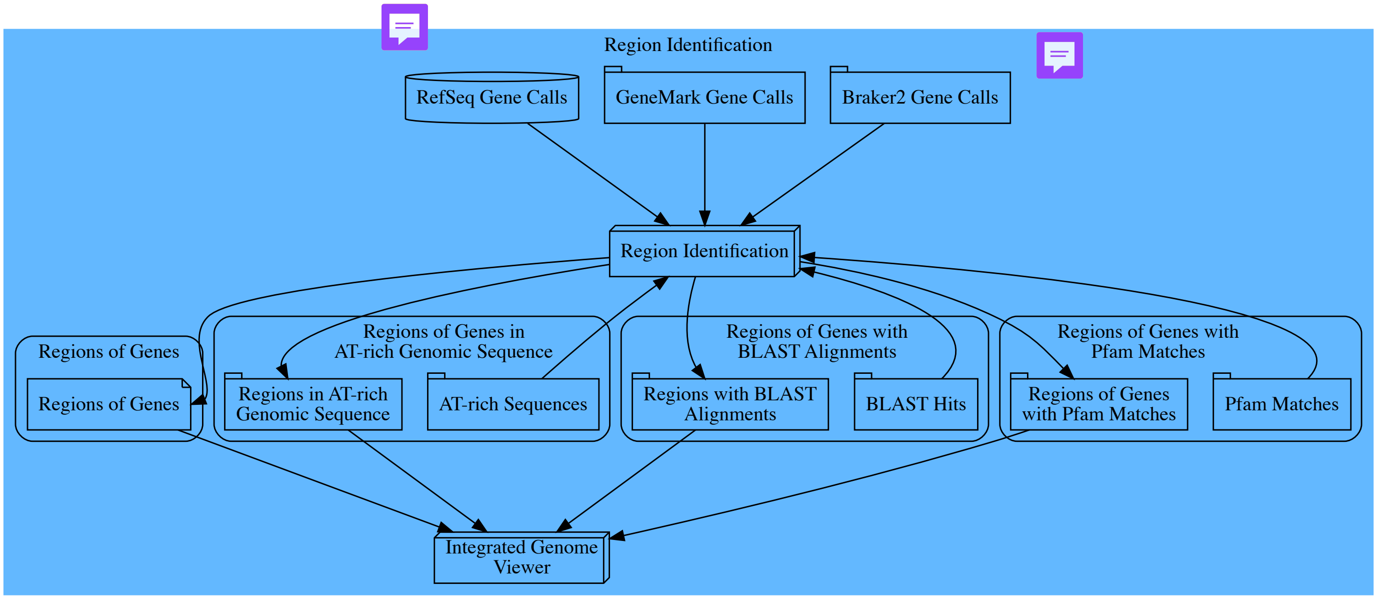

Figure 4.5: An overview of the region identification process for different datasets. Sections of the graph are outlined with rectangles, indicating that the datasets within were processed along with gene calls and discussed in their own results sections. Processing steps are represented by boxes, datasets of interest that were produced by this work are represented by folders, and datasets that were not produced by this work are represented by cylinders. Directed edges indicate the flow of data.

generated datasets. These additional datasets may contain other information such as ~~blast~~ alignments or functional annotations and can be included as features in the region identification process. Datasets outlined with rectangles indicate that the data within were processed in the context of the main gene calls and are discussed in their own results sections. Datasets output by the region identification process were then visualized with the Integrative Genomics Viewer, described in Section 4.16

4.14 Region Identification Algorithm

Predictions and results from 4.6 and 4.7 were processed into regions of overlapping gene predictions and annotations. A region is defined as a pair of genomic coordinates containing one start and one stop coordinate, that contains one or more overlapping features. We define a feature as a pair genomic coordinates containing one start and one stop coordinate that associated with a 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. Pseudo-code for the algorithm is shown in Algorithm 1. This processing was performed with Python v3.12.4[13] and the GFF

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
$sortedFeatures \leftarrow sortStartCoord(allFeatures)$
$regionList \leftarrow []$
$firstFeature \leftarrow firstItem(sortedFeatures)$
$lastFeature \leftarrow lastItem(sortedFeatures)$
for <each feature in sortedFeatures> do
    if $feature = firstFeature$ then
        $currentRegion \leftarrow newRegion(feature)$
        $currentRegion.start \leftarrow feature.start$
        $currentRegion.end \leftarrow feature.end$
    else if $feature = lastFeature$ then
        $regionList.append(currentRegion)$
        $return\ regionList$
    else if $feature\ overlaps\ currentRegion$ then
        $currentRegion.updatePositions(currentRegion, feature)$
        $currentRegion.addFeature(feature)$
    else
        $regionList.append(currentRegion)$
        $currentRegion \leftarrow newRegion(feature)$
        $currentRegion \leftarrow feature.start$
        $currentRegion \leftarrow feature.end$
    end if
end for
$return\ regionList$
```

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] Mosè Manni, Matthew R Berkeley, Mathieu Seppey, and Evgeny M Zdobnov. BUSCO: Assessing Genomic Data Quality and Beyond. *Current Protocols*, 1(12):e323, 2021.

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

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

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

[25] NCBI. NCBI SRA Toolkit, 2025.

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

[27] 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.

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

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

[30] Anthony Rhoads and Kin Fai Au. PacBio Sequencing and Its Applications. *Genomics, proteomics and bioinformatics*, 13(5):278–289, oct 2015.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.