UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOINFORMÁTICA

GERALDO CESAR CANTELLI

SOLUÇÃO DE INTEGRAÇÃO E AVALIAÇÃO DE SOFTWARE DE ANOTAÇÃO GENÔMICA EM *COFFEA SPP*

DISSERTAÇÃO – MESTRADO

CORNÉLIO PROCÓPIO

2020

GERALDO CESAR CANTELLI

SOLUÇÃO DE INTEGRAÇÃO E AVALIAÇÃO DE SOFTWARE DE ANOTAÇÃO GENÔMICA EM *COFFEA SPP*

Proposta de dissertação de mestrado apresentada ao Programa de Pós-Graduação em BioInformática da Universidade Tecnológica Federal do Paraná – UTFPR como requisito parcial para a obtenção do título de “Mestre em BioInformática”.

Orientador: Luiz Filipe Protasio Pereira

Co-orientador: Fabrício Martins Lopes

CORNÉLIO PROCÓPIO

2020

Dados Internacionais de Catalogação na Publicação

C229 Cantelli, Geraldo Cesar

Solução de integração e avaliação de software de anotação genômica em Coffea SPP / Geraldo Cesar Cantelli. – 2020.
116 f. : il. color. ; 31 cm.

Orientador: Luiz Filipe Protasio Pereira.
Coorientador: Fabrício Martins Lopes.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Bioinformática. Cornélio Procópio, 2020.
Bibliografia: p. 60-64.

1. Genoma. 2. Café. 3. Software - Desenvolvimento. 4. Bioinformática – Dissertações. I. Pereira, Luiz Filipe Protasio, orient. II. Lopes, Fabrício Martins, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Bioinformática. IV. Título.

CDD (22. ed.) 572.80285

Biblioteca da UTFPR - Câmpus Cornélio Procópio
Bibliotecário/Documentalista responsável:
Romeu Righetti de Araujo – CRB-9/1676

TERMO DE APROVAÇÃO

Geraldo Cesar Cantelli

SOLUÇÃO DE INTEGRAÇÃO E AVALIAÇÃO DE SOFTWARE DE ANOTAÇÃO GENÔMICA EM *COFFEA SPP*

Esta dissertação foi apresentada às 8 horas e 30 minutos de 04 de setembro de 2020 como requisito parcial para a obtenção do título de MESTRE EM BIOINFORMÁTICA, linha de pesquisa em Biologia Computacional e Sistêmica, Programa de Pós-Graduação em Bioinformática. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo citados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Luiz Filipe Protasio Pereira
Embrapa Café

André Yoshiaki Kashiwabara
UTFPR Câmpus Cornélio Procópio

Suzana Tiemi Ivamoto-Suzuki
UEL

Dedico este trabalho a pessoa de meus queridos pais, que tudo fizeram para o desenvolvimento da minha vida; e a mulheres também muito especiais no meu coração:

* Arnaldo Cantelli
* Domingas Ferreira Cantelli
✓ Ana Paula Cantelli
✓ Luciana Isidio Oliveira Cantelli

AGRADECIMENTOS

Agradeço inicialmente a Deus por tudo, ao orientador professor doutor Luiz Filipe Protasio Pereira, ao co-orientador professor doutor Fabrício Martins Lopes e ao professor doutor André Yoshiaki Kashiwabara pela oportunidade e visão de um tema tão interessante e pelas orientações durante a elaboração do trabalho. A Dra. Suzana Tiemi Ivamoto-Suzuki, Professora do Departamento de Agronomia da Universidade Estadual de Londrina (UEL), a Daniel Ribeiro de Brito, Mestre em Genética e Biologia Molecular (UEL) e a Leandro Carrijo Cintra, Técnico Responsável da Embrapa. Com relação às orientações quanto ao tratamento estatístico dos dados, o agradecimento pertence ao Dr. Marcelo Tutia, Professor da Faculdade de Tecnologia de Ourinhos e com relação à sugestão de métodos de avaliação de anotação de genomas, gratidão à Brian Haas, autor do software *PASA* e de vários artigos citados nesta obra. Agradecimentos à empresa Embrapa em geral pela possibilidade de utilização do Cluster de Computação. Também deixo aqui um muito obrigado a toda a equipe de professores da Universidade Tecnológica Federal do Paraná Câmpus Cornélio Procópio pelos ensinamentos ministrados e aos vários funcionários que colaboraram conjuntamente para que tudo fosse possível.

Pedi, e dar-se-vos-á; buscai, e achareis; batei, e abrir-se-vos-á. Porque todo o que pede, recebe; e o que busca, acha; e a quem bate, abrir-se-á. Ou qual de vós, porventura, é o homem que, se seu filho lhe pedir pão, lhe dará uma pedra? Ou, porventura, se lhe pedir um peixe, lhe dará uma serpente. Pois se vós outros, sendo maus, sabeis dar boas dádivas a vossos filhos, quanto mais vosso Pai, que está nos Céus, dará boas dádivas aos que lhe pedirem. (Mateus, VII: 7-11).

RESUMO

CANTELLI, Geraldo Cesar. SOLUÇÃO DE INTEGRAÇÃO E AVALIAÇÃO DE SOFTWARE DE ANOTAÇÃO GENÔMICA EM *COFFEA SPP* . 116 f. Dissertação – Mestrado – Programa de Pós-graduação em BioInformática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

Um dos maiores desafios da bioinformática é a análise de genomas completos, por exemplo, a identificação de genes preditos computacionalmente e a sua associação com as respectivas funções biológicas. Portanto é importante o design de experimentos que possam testar essas previsões e compará-las com outras já existentes para que se possa mensurar seu desempenho. Realizar o experimento apenas com um software não seria interessante pela necessidade de comparar algoritmos e sua eficiência. Devido ao volume crescente de dados genômicos e transcriptômicos disponíveis, são necessárias pipelines eficientes e acessíveis para gerar predições gênicas e inferir com um maior grau de confiabilidade as suas respectivas funções biológicas. Como melhorar a qualidade da anotação genômica, evitando "over" ou "under prediction" e obtendo mais precisão? Neste trabalho, estudamos qual característica é mais interessante para um software de anotação genômica comparando dois programas, *PASA* e *MAKER*, analisando o genoma de *Coffea canephora*, *C. eugenioides* e *C. arabica*. Através da realização dessas pipelines, notou-se através de programas como BUSCO e Quast um aprimoramento no genoma das amostras de café e realizada uma comparação estatística entre esses dois programas. Além disso é proposta uma nova ferramenta automatizada que permite repetir algumas das análises realizadas neste trabalho. Os resultados mostram a eficácia do uso da detecção de todas as possibilidades de splices alternativos no algoritmo de anotação, devido o *PASA* encontrar mais genes exclusivos e genes localizados igualmente em diferentes regiões dos cromossomos, o que é difícil para muitos preditores de genes. Foram geradas novas versões da anotações dos genomas de *C. arabica*, *C. canephora* e *C. eugenioides* para que possam ser disponibilizadas para utilização pela comunidade científica. Foi desenvolvido um programa Ensemble Solution para viabilizar a automatização da avaliação de software de anotação genômica, o qual trabalhando com arquivos de GFF3, produz listas de genes encontrados exclusivamente por cada software avaliado e gera diagramas de Venn, permitindo importar dados do GenBank (como a tradução das proteínas) e gerar relatórios mais completos.

Palavras-chave: *Coffea canephora*, *Coffea arabia*,*Coffea eugenioides*, *PASA*,

MAKER, Mann-Whitney, ...

ABSTRACT

CANTELLI, Geraldo Cesar. ENSEMBLE SOLUTION AND EVALUATION OF GENOMIC ANNOTATION SOFTWARE IN *COFFEA SPP*. 116 f. Dissertação – Mestrado – Programa de Pós-graduação em BioInformática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2020.

One of the biggest challenges of bioinformatics is the analysis of complete genomes, for instance the identification of computationally predicted genes and its association to respective biological functions. Therefore, it is important to design experiments that can test these predictions and compare them with existing ones so that you can measure their performance. With a growing volume of genomic and transcriptomic available data, efficient and affordable pipelines to perform a good gene annotation process are needed. How to improve the correct genome annotation avoiding over or under prediction to obtain more accurary? In this work we study which characteristic is more interesting to a genomic annotation software comparing two software, *PASA* and *MAKER*, analyzing the genome of *Coffea canephora*, *C. eugenioides* and *C. arabica*. We also executed a quality improvement in these *Coffea* genome annotation and peformed statistical comparison between these two software. Besides it is proposed an automated tool which allows to repeat some of the analyses performed in this work. Results show the effectiveness of using detection of all alternative splicing possibilities in the algorithm of annotation due to *PASA* finding more exclusive genes (compared with *MAKER*) and located genes equally in different regions of the chromosomes, which is difficult for many gene predictors. New versions of the annotation of the genomes of *C. arabica*, *C. canephora* and *C. eugenioides* were generated to be made available for use by the scientific community. The Ensemble Solution program was developed to make possible evaluation of genomic annotation software, GFF3 files, lists of genes exclusively and Venn diagrams, to import GenBank properties and generate more complete reports.

Keywords: *Coffea canephora*, *Coffea arabia*,*Coffea eugenioides*, *PASA*, *MAKER*, Mann-Whitney, ...

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE CÓDIGOS

LISTA DE SIGLAS

APD	Algoritmo de Programação Dinâmica
AUC	Area under the ROC curve
BUSCO	Benchmarking Universal Single- Copy Orthologs
cDNA	DNA complementar
CDS	Região que codifica proteína
CECAFÉ	Conselho dos Exportadores de Café do Brasil
CGH	Coffee Genome Hub
DNA	Ácido desoxirribonucleico
EST	Expressed sequence tags
FL-cDNA	Full-length cDNA
FPR	False positive rate
GNU	acrônimo: GNU´s not UNIX
mRNA	RNA mensageiro
NCBI	National Center for Biotechnology Information
ORF	Open Reading Frames
PASA	Program to Assemble Spliced Alignments
PDF	Portable Document Format, formato de arquivo criado pela empresa Adobe Systems
RNA	Ácido ribonucleico
RNA-Seq	RNA Sequencing
ROC	(Receiver ou Relative) Operating Characteristic
rRNA	RNA ribossômico
SRA	Sequence Read Archive
TPR	True positive rate
tRNA	RNA transportador
UTR	Untranslated region

SUMÁRIO

1 INTRODUÇÃO

Entre os tipos de dados biológicos, os dados genômicos são analisados pelas ferramentas de bioinformática tais como preditores de genes, analisadores de sequência e software voltados para similaridade. Por essa razão, o uso de programas gerais e orientados tem aumentado na última década (MORENO *et al.*, 2018). O genoma completo de várias espécies está armazenado e disponível por exemplo em arquivos de formato FASTA (PEARSON, 2016), disponíveis no site do *National Center for Biotechnology Information* (JENUTH, 2000), constituindo uma vasta fonte de dados prontos para serem processados e tornarem-se informação útil. O conhecimento relacionado a genomas tem múltiplas aplicações, as quais incluem a indústria farmacêutica, através da descoberta de novas drogas, ou na agricultura, pelo desenvolvimento de novos cultivares. Entretanto, muitos desafios têm sido relatados quanto aos programas para a anotação genômica "a anotação automatizada de grandes '*drafts*' de genomas fragmentados ainda é muito difícil e com muitos erros, além de possui contaminações em assemblies de *drafts* levam a erros na anotação que tendem a se propagar através das espécies" (SALZBERG, 2019).

Ainda de acordo com SALZBERG (2019), um pipeline de anotação tal como o *MAKER* (CANTAREL *et al.*, 2008) pode usar dados de RNA-seq, combinado com bancos de dados de proteínas conhecidas e outras entradas, como dados de *Repeat Mask* e genoma no formato FASTA. Essa combinação é realizada para encontrar genes e até atribuir uma possível nomenclatura. Contudo RNA-seq não captura todos os genes em um genoma e *drafts* de genoma podem conter milhares de contigs desconexos, muitos genes poderão estar divididos entre vários contigs (ou scaffolds) cuja ordem e orientação são

desconhecidas.

Para YANDELL; ENCE (2012), embora o sequenciamento tenha se tornado mais fácil, de certa forma, a anotação genômica tornou-se mais desafiadora.

Então há um problema de como melhorar a qualidade evitando predições superestimando e subestimando o número de genes (*over ou under prediction*) e dessa forma, obter maior precisão na identificação de genes. Para encontrar a resposta a essa pergunta este trabalho adota análises na performance de dois programas: *PASA* (HAAS *et al.*, 2008) e *MAKER*, e discutimos os resultados frente a suas propriedades estruturais e funcionais.

Uma importante propriedade é a predição dos *splicing* alternativos e essa função está inclusa no software *PASA* que usa essa informação para corrigir erros na estrutura criada por preditores de genes *ab initio* e aliado à técnica de predição por detecção de sequências homólogas, este software propõe grande precisão na geração de arquivos GFF3 que descrevem como os genes deveriam ser após o pipeline ser executado (HAAS *et al.*, 2003).

Neste trabalho foi realizada uma análise estatística comparativa entre os resultados obtidos pelas pipelines de anotação genômica do *PASA* e do *MAKER*, usando dados genômicos de café para determinar: primeiro se houve uma melhoria em suas anotações e, em segundo lugar, qual software obteve melhor desempenho. Essas análises envolveram informações dos programas BUSCO v3.0.2 (SIMÃO *et al.*, 2015), Quast v4.6.1 (GUREVICH *et al.*, 2013) e GFFCompare (PERTEA; PERTEA, 2020). Também foi desenvolvido um novo software, que permite que uma comparação como essa seja repetida de maneira semi-automatica: *the Ensemble Solution and Evaluation of Genomic Annotation Software*, escrita em Perl, que gera um gráfico e um histograma, ambos de Venn, e listas de genes encontrados exclusivamente por cada pipeline.

Para SIMÃO *et al.* (2015), a medida de completude da montagem é importante pois afeta a interpretação dos dados e ajuda a orientar melhores estratégias de montagem e anotação. Com o crescente número de genomas

sequenciados disponíveis, o conhecimento de seu conteúdo gênico está se consolidando e pode ser usado para desenvolver uma medida evolutiva da integridade do genoma (os *Benchmarking Universal Single Copy Orthologs*, ...BUSCO. Assim, deve-se notar que, embora as avaliações do BUSCO visem estimar com robustez a integridade dos conjuntos de dados, as limitações técnicas (particularmente a predição de genes) podem inflar proporções de BUSCOs "fragmentados" e "ausentes", especialmente para genomas grandes.

Este trabalho ainda relaciona a comparação do desempenho das pipelines com seu *modus operandi*, investigando a razão dos resultados especialmente particulares de um deles com suas funcionalidades específicas. Além disso, as anotações genômicas de três espécies de café de (*C. arabica*, *C. canephora* e *C. eugenioides*) foram aprimoradas e essas mudanças disponibilizadas em arquivos FASTA.

2 REVISÃO DE LITERATURA

2.1 A IMPORTÂNCIA DO CAFÉ

O gênero *Coffea* tem 124 espécies, mas apenas o alotetraplóide *Coffea arabica L.* e o diplóide *Coffea canephora* apresentam importância econômica, são responsáveis por aproximadamente 60% e 40% da produção mundial de café, respectivamente. Segundo o Ministério da Agricultura, Pecuária e Abastecimento, o Brasil é o maior produtor e exportador de café e o segundo maior consumidor da bebida no mundo. É o quinto produto do conjunto de exportação brasileira, movimentando US$ 5,2 bilhões em 2017. (ABASTECIMENTO, 2020).

De acordo com a Associação Nacional do Café dos Estados Unidos da América, a atividade econômica gerada em torno do café naquele país atinge 1,6% de seu produto interno bruto e em 2015 chegou a $ 225.2 bilhões, sendo ainda responsável por 1.694.710 empregos (NCA, 2019). Os Estados Unidos é o principal comprador do café brasileiro.

Para demonstrar a importância do mercado mundial de café e a respectiva participação brasileira, tem-se a Figura 2.1:

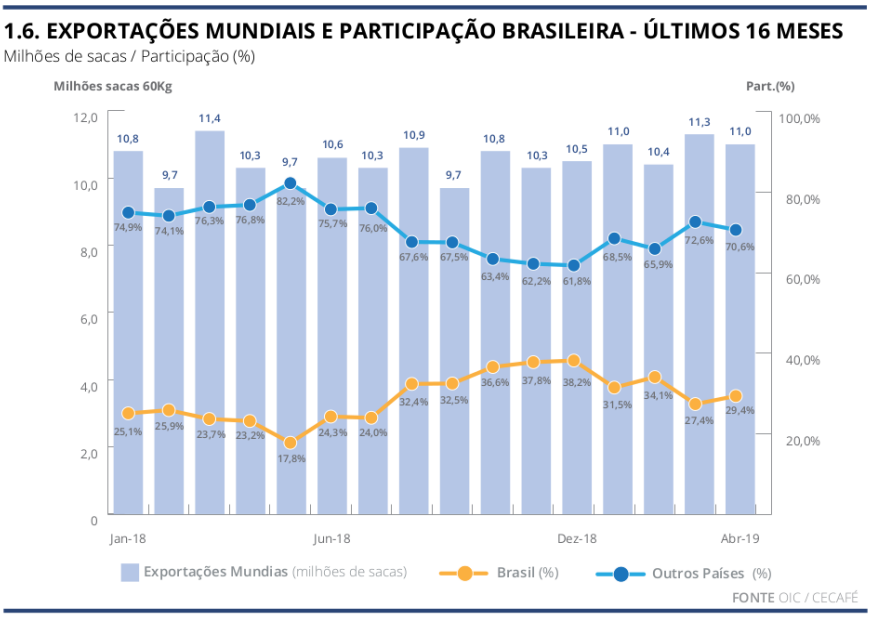

Figura 2.1: Exportações de Café Mundial - Participação Brasileira

Fonte: Cecafé - Setembro 2019

O genoma de *C. canephora* e *C. arabica* foram publicados recentemente. A anotação genômica de *C. canephora* está disponível no site Coffee Genome Hub (DEREEPER *et al.*, 2014), já no site do National Center for Biotechnology Information (JENUTH, 2000) temos os dados das três espécies estudadas neste trabalho: *C. arabica*, *C. eugenioides* e *C. canephora*. Em artigo publicado por VIEIRA *et al.* (2006) é apresentado o Projeto Genoma Café, recursos genômicos baseados em sequenciamento de *Expressed Sequence Tags (ESTs)*, cujos dados podem ser encontrados em http://www.lge.ibi.unicamp.br/cafe.

É importante refazer a anotação genomica do *C. arabica* pois ele é um alopoliploide, portanto ocorre uma maior dificuldade de ter um boa montagem genômica, levando a maior dificuldade de anotação e impactando a qualidade da mesma.

2.2 ASPECTOS DE ANOTAÇÃO GENÔMICA

Vários arquivos de entrada de dados utilizados nesse trabalho estão no formato FASTA. Este é o nome de um pacote de software para alinhamento de seqüências de DNA e proteínas descrito primeiramente como FASTP por David J. Lipman e William R. Pearson em 1985 (PEARSON; LIPMAN, 1985). O formato de arquivo definido pelo FASTA é um dos principais formatos utilizados para armazenamento de sequências biológicas.

Sobre o aspecto técnico, a anotação genômica e não o sequenciamento genômico é o grande gargalo da bioinformática atual pois se sequencia mais dados do que se é possível analisar, segundo CANTAREL *et al.* (2008). Seu artigo cita que Eucariontes são um caso particularmente complicado por seu grande tamanho de informação e por haverem partes de genes contidas em *introns* o que torna difícil a anotação. Afirma ainda que a anotação e a distribuição desses genomas pode beneficiar a comunidade biomédica, mas que isso esbarra nas dificuldades para muitas dessas comunidades onde falta *expertise* em bioinfo. A solução seria publicar não só os dados mas também ferramentas de anotação genômica online para que outros grupos possam colaborar com os esforços científicos e para tanto essas ferramentas precisam ser mais simples e mais portáveis, como por exemplo para usuários não tão avançados de sistemas UNIX e Linux (padrão POSIX).

Segundo CANTAREL *et al.* (2008), apesar dos melhores esforços da comunidade bioinformata, o grande número de genomas não anotados continua a crescer, revelando a necessidade urgente de pipelines mais simples e portáveis.

De acordo com ARMSTRONG *et al.* (2019), a tarefa de criar anotações de montagens de genomas automaticamente começou a ser estudada desde o surgimento do primeiro genoma completo (*full-length*) em meados dos anos 1990s (LETOVSKY *et al.*, 1998; LUKASHIN; BORODOVSKY, 1998; KULP *et al.*, 1996). Esta tarefa é frequentemente dividida em duas categorias:

- predição *ab initio*, ou predição computacional de estruturas *exon-intron*

usando modelos estatísticos;

- abordagens baseadas em alinhamento de sequências, com mapeamento de toda *expressed sequence tag* (EST), *complementary DNA* (cDNA), ou sequências de proteínas para dentro de montagens de sequências para descobrir transcritos.

Em PRUITT *et al.* (2006) e CANTAREL *et al.* (2008) nota-se que algumas pipelines de anotação combinam ambas fontes de predição de transcritos para gerar um conjunto final de anotação.

Corroborando com essas definições, YANDELL; ENCE (2012) coloca que a anotação genômica das estruturas dos genes é dividida em duas fases distintas. Na primeira, que seria a fase da computação, *expressed sequence tags (ESTs)*, proteínas, e assim por diante, são alinhadas ao genoma são geradas predições de genes *ab initio* e/ou orientadas a evidências. Já na segunda fase esses dados são sintetizados em anotação genômica; esse processo envolve muitas e diferentes ferramentas e os programas que computam os dados (evid ências) e os usam para criar anotações genômicas são chamados de pipelines de anotação. A Figura 2.2 exemplifica esse processo:

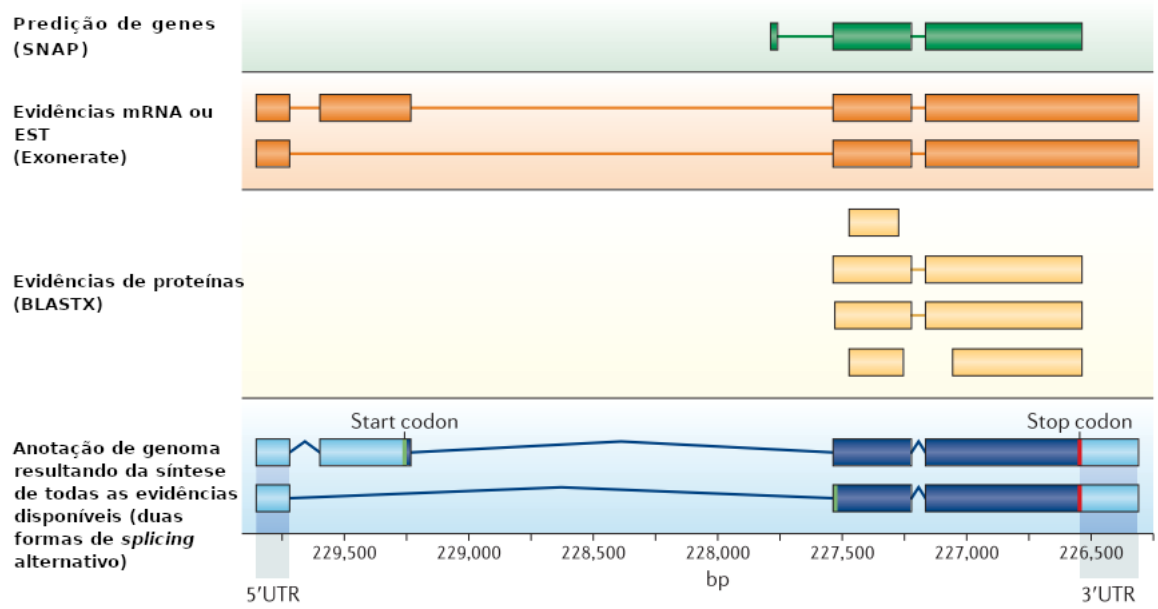

Figura 2.2: Predição de Genes versus Anotação de Genes

Fonte: (YANDELL; ENCE, 2012)

Um estudo publicado por HAAS *et al.* (2011) também evidencia que há duas categorias de programas preditores de genes: predição de gene *ab initio* e por detecção de sequências homólogas; o texto sugere a junção de ambas as técnicas para se obter um melhor resultado e para tanto podem-se utilizar sequências transcritas homólogas para, através do efeito de comparação pelo software *PASA*, chegar à determinação de informações precisas sobre *introns, exons*, enfim dos próprios genes estudados. Argumenta ainda o autor que preditores de genes baseados em sequências homólogas são considerados uma forte tendência para localização e modelagem de estruturas de genes quando dados experimentalmente verificados já estão disponíveis ou quando padrões conservados podem ser inferidos de alinhamentos de genomas de espécies relacionadas e que estes alinhamentos unidos provêm forte evidência para compor estruturas de genes, em muitos casos resolvendo completamente *exons* e *introns* e até mesmo revelando a localização de um gene candidato. Reforça ainda que sequências transcritas, quando derivadas do mesmo organismo cujo genoma está sendo sequenciado, provêm a mais acurada forma de

evidência na resolução da estrutura de genes, pois delineiam precisamente as fronteiras de *introns* e *exons*. Esses transcritos incluem *expressed sequence tags* (ESTs), *full-length cDNAs* (FL-cDNAs), e mais recentemente , sequências de cDNA derivadas do sequenciamento de nova geração de transcriptoma de sequências curtas (chamado RNA-Seq). Ferramentas como *PASA*, EST-Genes (EYRAS *et al.*, 2004), e CallReferenceGenes (MCGUIRE *et al.*, 2008) montam múltiplos alinhamentos sobrepostos de cDNA em mais estruturas *full-length* de genes. Essas ferramentas são capazes de gerar múltiplos modelos de transcritos por gene quando estes diferem em alinhamentos sobrepostos gerados por *splicing* alternativo.

"ESTs e cDNAs têm sido ferramentas poderosas na descoberta de genes e estudos de expressão gênica desde que sequências completas de genoma são uma realidade" (ADAMS *et al.*, 1991 apud HAAS *et al.*, 2003). "Alinhamentos de *full-length cDNAs (FL-cDNAs)* tem provado serem muito úteis em resolver estruturas de genes e melhorar anotações em Arabidopsis" (SEKI *et al.*, 2002 apud HAAS *et al.*, 2003).

A demonstração da pipeline de montagem de alinhamentos e anotação do *PASA* encontra-se neste trabalho através de todo o Apêndice A. Já a respeito do software *MAKER*, foi originalmente desenvolvido para anotações *de novo* de modelos de organismos mais recentes e que depois expandiu-se para um multiuso de anotação de genomas e ferramenta de curadoria (HOLT; YANDELL, 2011). Segundo CAMPBELL *et al.* (2014), além de trabalhar com anotações *de novo*, pode ser utilizado para atualizar anotações existentes à luz de novas evidências experimentais e para controle de qualidade de modelos de genes produzidos por outras pipelines de anotação.

O *MAKER* usa o CGL (YANDELL *et al.*, 2006) commom object model, que estende as classes GenericHit e GenericHSP do Bioperl (*http://www.bioperl.org*) com métodos que facilitam análises comparativas e anotações automáticas. A construção modular do *MAKER* permite dividir o processo de anotação em uma série de cinco atividades discretas que são facilmente interoperáveis: computação, filtro/cluster, polimento, síntese e anotação (Fi-

gura 2.3). *MAKER* executa essas ações em sequências de qualquer tamanho, cortando automaticamente a sequência de entrada em séries de pedaços (o padrão é 100 kb), executando cada cálculo e integrando os resultados.

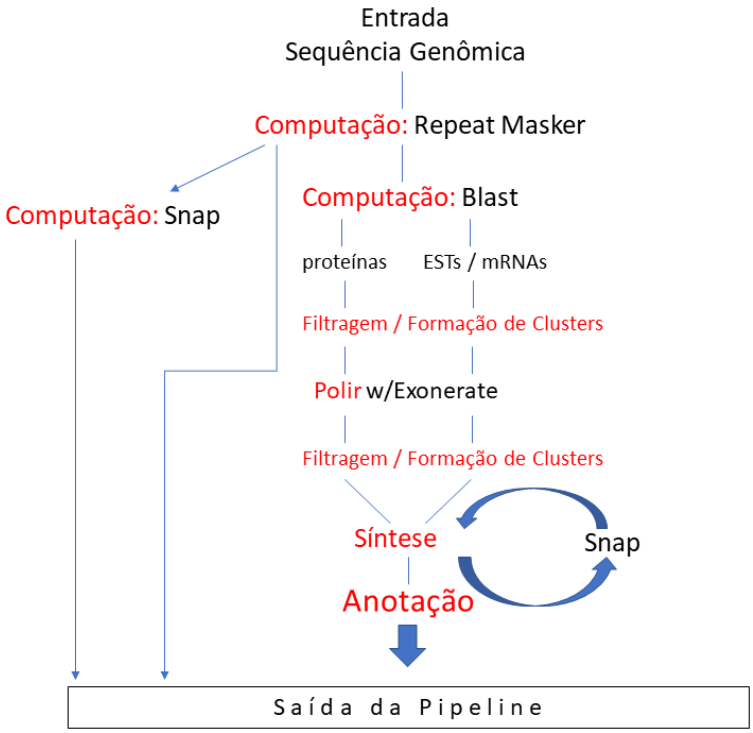

Figura 2.3: *MAKER Pipeline*

Fonte: (CANTAREL *et al.*, 2008)

A Figura 2.3 exibe um diagrama da dinâmica de funcionamento da pipeline do *MAKER*. Sua configuração padrão usa quatro programas externos: RepeatMasker (http://repeatmasker.org), BLAST (SIMÃO *et al.*, 2015), Exonerate (SLATER; BIRNEY, 2005), and SNAP (KORF, 2004). A menos que as repeats sejam efetivamente mascaradas, a predição e a anotação genômicas irão conter porções de transposons e vírus. O RepeatMasker é utilizado para procurar dentro do genoma por repeats de baixa-complexidade, então elas são "soft-masked" (transformadas em letras minúsculas). Exonerate entra em ação para realinhar sequências (normalmente de *ESTs* e *mRNAs*), que em seguida passam por processos de filtragem e formação de *clusters*. SNAP é

um software que pretende realizar predição de genes *ab initio*. Ao final do processo todo, a anotação é gerada (CANTAREL *et al.*, 2008).

Ainda com relação ao *MAKER*, a pipeline poderia incluir Repeat Masker, proteínas e entrada de RNA-seq mas essas informações não foram utilizadas nesse trabalho. Os parâmetros de utilização deste software encontram-se no Apêndice B.2.

2.3 ALGORITMO DE MONTAGEM DE ALINHAMENTOS

O algoritmo de programação dinâmica chamado *Program to Assemble Spliced Alignments (PASA)* é na realidade uma ferramenta para montagem de alinhamento de cDNA e identificação de variantes de *splicing* alternativo que foi montado e descrito por HAAS *et al.* (2003) inicialmente para consolidar e maximizar as *untranslated regions* (UTRs) de *FL-cDNAs* de Arabidopsis e integrar evidências de ESTs em regiões onde nenhum alinhamento de FL-cDNA estivesse disponível, provendo o máximo de substratos para a atualização e incremento da anotação genômica da Arabidopsis. Ainda segundo o estudo, o objetivo do algoritmo de montagem é encontrar, para cada alinhamento *a*, o maior *assembly* (termo em inglês para montagem) que contém *a*, por exemplo, o *assembly* contendo *a* com o máximo número de outros ESTs e cDNAs.

O *maximal assembly* de alinhamentos é usado como substrato para a criação de modelos de genes ou para a modificação de modelos de genes existentes e consiste do *assembly* com o maior número de alinhamentos compatíveis. Alinhamentos compatíveis são aqueles cujas regiões sobrepostas são idênticas e têm mesma orientação. Por exemplo na Figura 2.4, o alinhamento A é compatível com o B; também o alinhamento B é compatível com o C. Contudo n ão acontece a propriedade transitiva pois o alinhamento A não é compatível com o C, logo os três não podem formar um *assembly*:

Figura 2.4: Exemplo de compatibilidade de alinhamentos

Fonte: (HAAS *et al.*, 2003)

A demonstração do funcionamento do algoritmo do *PASA* encontra-se também em HAAS *et al.* (2003), que para simplificar consideram apenas uma fita da sequência genômica e a trata como uma linha marcada com inteiros, com os menores valores à esquerda. Também usam o termo cDNA para se referir coletivamente tanto a FL-cDNA quando a ESTs. O *span* de um alinhamento é definido como a faixa compreendida do seu início até seu fim. Veja um exemplo de constituição dos assemblies na pipeline do software PASA na Figura 2.5:

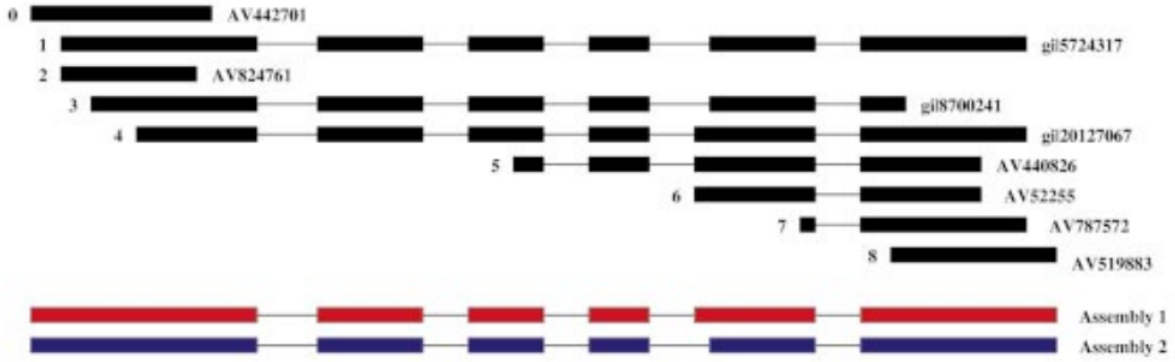

Figura 2.5: Construção dos assemblies no PASA

Fonte: (HAAS *et al.*, 2003)

Nela vê-se que os dois assemblies abaixo puderam ser constituídos graças a ação do *splicing* alternativo percebido pelo próprio software PASA e representado através dos alinhamentos.

O cálculo do *maximal assembly* ocorre por programação dinâmica.

Primeiro, todos os alinhamentos são ordenados ascendentemente a partir de suas posições iniciais. Depois, cada par de alinhamentos que se sobrepõe é testado para compatibilidade. Todos os cDNAs sobrepostos num *assembly* devem ser compatíveis.

Considere L_a o máximo número de cDNAs em um *assembly* contiguo que termina em um alinhamento *a*, e que inclui *a*, com alinhamentos compatíveis contidos no *span* de *a* e alinhamentos que terminam estritamente antes do fim de *a*, mas sem alinhamentos que estritamente contém *a*.

Para alinhamentos sobrepostos compatíveis *a* e *b*, considere $C_{a/b}$ ser o número de alinhamentos compatíveis com *a* contidos no *span* de *a* (incluindo ele próprio) porém não contidos em *b*. E seja C_a o número de alinhamentos compatíveis com *a* contidos no *span* de *a*.

Então tem-se:

$$L_a = \max_b \left\{ C_a,\ L_b + C_{a\setminus b} \;\middle|\; \begin{array}{l} \textbf{\textit{b é compatível com a,}} \\ \textbf{\textit{b é orientado a esquerda de a,}} \\ \textbf{\textit{a não está contido dentro de b}} \end{array} \right\}$$

Figura 2.6: Cálculo de *La*

Fonte: (HAAS *et al.*, 2003)

Os valores de C_a são identificados quando os alinhamentos sobrepostos forem testados para compatibilidade. Destas listas de alinhamentos, os valores de $C_{a/b}$ podem ser definidos por operações de mesclagem (*merge operations*).

Durante esse processamento, cada alinhamento *a* mantém um ponteiro p_a para o alinhamento *b* que atinge o maior número de alinhamentos compatíveis, que define L_a.

O maior de todos os valores de L_a, L_{a^*}, representa o maior número de alinhamentos montados, ou seja, o *assembly* contendo o maior número

de cDNAs compatíveis. Começando pelo alinhamento *a**, rastreando de trás para frente os ponteiros p_a junto com os alinhamentos de $C_{a/b}$, resultará nos cDNAs que compõe o *maximal assembly*.

Se quaisquer alinhamentos não estão incluídos no *maximal assembly*, então existem alinhamentos conflitantes, indicando *isoforms de splicing* alternativo ou transcritos sobrepostos correspondentes a genes diferentes; alinhamentos pareados incompatíveis de alternativos sites acessores ou doadores, *unspliced introns*, *skipped exons*, ou orientações de *splice* opostas.

Considere *a'* um cDNA não incluso no *maximal assembly*. Rastreando de trás para frente os ponteiros p_a obtém-se o maior *assembly* à esquerda de *a'* mas infelizmente estes ponteiros não provêm o maior *assembly* à direita de *a'*, para tanto usa-se a fórmula:

$$R_a = \max_b \left\{ C_a,\ R_b + C_{a\setminus b} \;\middle|\; \begin{array}{l} \textbf{\textit{b é compatível com a,}} \\ \textbf{\textit{b é orientado a esquerda de a,}} \\ \textbf{\textit{a não está contido dentro de b}} \end{array} \right\}$$

Figura 2.7: Cálculo de *Ra*

Fonte: (HAAS *et al.*, 2003)

Onde R_a representa o máximo número de cDNAs em um *assembly* contendo *a* consistindo somente de cDNAs que iniciam à direita e terminam em *a*. Como antes, cria-se um ponteiro q_a para o alinhamento *b* que atinge o valor para *maximal* do conjunto.

O total número de cDNAs no maior *assembly* contendo um alinhamento qualquer *a* é igual a:

$$\max_b \{L_b + R_b - C_b | \text{b contém a}\} \qquad (14)$$

É necessário retirar C_b pois estes cDNAs já foram contados em ambos

L_b e R_b. Os cDNAs que compõe este *maximal assembly* podem ser obtidos rastreando os ponteiros p_x e q_x e incluindo os correspondentes cDNAS $C_{x/y}$. O algoritmo do *PASA* continua buscando todos os alinhamentos ainda não incluídos no *maximal assembly* que tenham o maior valor para a fórmula 14. O *assembly* correspondente (com os novos alinhamentos que contém) é então adicionado à coleção de *maximal assemblies* e o processo é repetido até que todos os alinhamentos tenham sido incluídos a pelo menos um *assembly*.

2.4 TESTE DE MANN-WHITNEY

O teste de Mann-Whitney foi desenvolvido primeiramente por F. Wilcoxon em 1945, para comparar tendências centrais de duas amostras independentes de tamanhos iguais. Em 1947, H.B. Mann e D.R. Whitney generalizaram a técnica para amostras de tamanhos diferentes (CARMO, 2019).

Ainda conforme CARMO (2019), quando se dispõe de uma amostra pequena e a variável numérica não apresenta sabidamente uma variação normal (ou é possível verificar satisfatoriamente), ou ainda, quando não há homogeneidade das variâncias (embora exista uma correção no teste t que considera as variâncias desiguais), o teste t não é apropriado. Para exemplificar uma situação onde o teste t acusaria falsamente uma associação estatisticamente significativa. Imagine que em um dos dois grupos se observe um valor muito discrepante, em função desse único valor, em sendo muito maior do que os outros, o grupo a que ele pertence apresentará uma média elevada, o que aumentará a estatística do teste t, com um consequente p-valor associado pequeno. Nessa situação, pode-se utilizar o teste não paramétrico de Mann-Whitney.

2.5 JUSTIFICATIVA DO EMPREGO DOS PROGRAMAS BUSCO E QUAST

NEUMANN (2019) realizou uma pesquisa sobre métodos e software para a avaliação de anotação genômica e BUSCO e Quast estão no topo da

lista, ordenada pelo número de citações no Google Acadêmico (atualizado em 03/09/2020).

Tabela 2.1: 10 melhores resultados da Pesquisa Sistemática ordenada pelo número de citações no Google Acadêmico

Autor	Título	Ano	Citações
SIMÃO *et al.*	*BUSCO: Assessing genome assembly and annotation completeness with single-copy orthologs*	2015	3645
GUREVICH *et al.*	*QUAST: Quality assessment tool for genome assemblies*	2013	2718
BRADNAM *et al.*	*Assemblathon 2: Evaluating de novo methods of genome assembly in three vertebrate species*	2013	596
EARL *et al.*	*Assemblathon 1: A competitive assessment of de novo short read assembly methods*	2011	511
HUNT *et al.*	*REAPR: A universal tool for genome assembly evaluation*	2013	350
PHILLIPPY *et al.*	*Genome assembly forensics: Finding the elusive mis-assembly*	2008	290
NARZISI; MISHRA	*Comparing de novo genome assembly: the long and short of it*	2011	161
RAHMAN; PACHTER	*CGAL: Computing genome assembly Pachter likelihoods*	2013	91
DARLING *et al.*	*Mauve assembly metrics*	2011	95

Fonte: (NEUMANN, 2019)

O software BUSCO apresenta 3645 citações desde 2005 e o QUAST, 2718 citações desde 2013. Esses programas foram considerados adequados para alimentar o processo da análise estatística proposta nesse trabalho por

esses indicadores, o primeiro na sua versão 3.0.2 e o segundo na versão 4.6.1.

3 OBJETIVOS

3.1 OBJETIVO GERAL

O presente trabalho propõe uma comparação de performance e eficiência entre os programas *MAKER* e *PASA* através da revisão da anotação dos genomas *Coffea*.

3.2 OBJETIVOS ESPECÍFICOS

- Analisar e descrever as diferenças entre as entradas e saídas das pipelines dos programas *MAKER* e *PASA*;
- Realizar o aprimoramento na qualidade da anotação genômica de *C. canephora*, *C. arabica* e *C. eugenioides*;
- Estudar qual característica leva a uma melhor montagem das contigs;
- Desenvolver e disponibilizar uma ferramenta automatizada para comparação de resultados de programas de anotação genômica Ensemble Solution;
- Publicar arquivos FASTA com as anotações das mudanças produzidas no genoma das três amostras de café;

4 MATERIAIS E MÉTODOS

4.1 PROCESSAMENTO DOS DADOS SEQUENCIADOS EM LARGA ESCALA

4.1.1 PREDIÇÃO DE GENES DE COFFEA SPP COM OS SOTWARES *PASA* E *MAKER*

Nesse trabalho foram adotados diferentes bancos de dados de genômica/transcriptômica para as três espécies de café: os bancos de dados dos sites *Coffea Genome Hub* (DEREEPER *et al.*, 2014) e NCBI (JENUTH, 2000) os transcritos fornecidos por (IVAMOTO *et al.*, 2017). Foram usados sempre os dados montados, não o *raw-data*.

Para *C. canephora* uma das fontes utilizadas foi o *assembly AUK_PRJEB4211_v1* do site do NCBI, com as seguintes características:

- *BioSample: SAMEA3146290*
- *BioProject: PRJEB4211*
- *GenBank assembly accession: GCA_900059795.1 (latest)*

A outra fonte para essa mesma espécie foi o *C. canephora genome v1.0* disponível no site do *Coffea Genome Hub*, curado manualmente e com a análise sob título *Whole Genome Assembly and Annotation of Coffea canephora (Genoscope)*.

Já para *C. eugenioides* a amostra foi obtida no site do NCBI com as seguintes especificações:

- *BioSample: SAMN10269643*

- *BioProject: PRJNA497891*
- *GenBank assembly accession: GCA_003713205.1 (latest)*

Com relação a *C. arabica*, o genoma e uma amostra de transcriptoma também tiveram como fonte o banco de dados do NCBI, submetido pela *Johns Hopkins University*:

- *BioSample: SAMN10272287*
- *BioProject: PRJNA497895*
- *GenBank assembly accession: GCA_003713225.1 (latest)*

E a outra amostra de transcriptoma de *C. arabica* foi fornecida pela Dra. Suzana Tiemi Ivamoto-Suzuki (IVAMOTO *et al.*, 2017).

4.2 PRÉ-PROCESSAMENTO

Inicialmente, é necessário retirar do conteúdo do arquivo de transcritos as informações que poderiam atrapalhar o processamento das pipelines. A pipeline do *PASA* utiliza por padrão o programa externo SeqClean para tal atividade. Após o banco de dados ser configurado (MySql ou SQLite), iniciamos a pipeline do *PASA* (todos os comandos envolvidos na pipeline completa estão no Apêndice A).

No caso do pipeline *MAKER*, é necessário criar três arquivos de configuração: maker_bopts.cpl, maker_exe.cpl and maker_opts.cpl. Entre os três arquivos criados pelo comando inicial, o único que foi aqui customizado foi o maker_opts.cpl (comandos e explicações a respeito dessa configuração podem ser vistas nos Apêndices B.1 e B.2).

4.3 COMPARAÇÕES INICIAIS

Este trabalho é iniciado com ambas as pipelines sendo executadas (Figura 4.1):

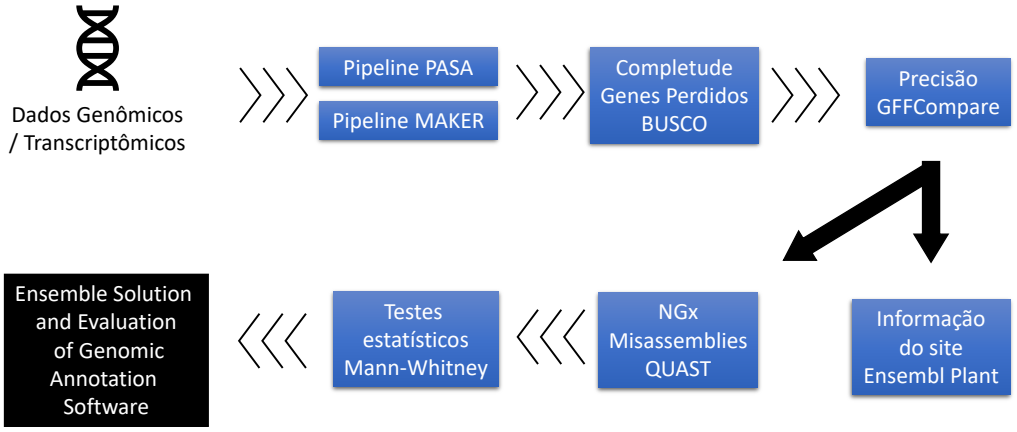

Figura 4.1: Diagrama mostrando como as análises comparativas foram conduzidas desde as pipelines

Fonte: O Autor

Depois de executar as pipelines, iniciou-se um processo estatístico de comparação para verificar qual pipeline foi mais eficiente. Os parâmetros utilizados foram o número de de genes ausentes do BUSCO, também seu índice de completude e mais a precisão calculada pelo software GFFCompare.

As métricas intuitivas para descrever genoma, conjunto de genes ou integridade do transcriptoma nos resultados do BUSCO foram: completo (C), duplicado (D), fragmentado (F), ausente (M) e número de genes utilizados (n). Os genes recuperados são classificados como “completos” quando seus comprimentos estão dentro de dois desvios padrões do grupo de comprimento médio do BUSCO. Os genes recuperados apenas parcialmente foram classificados como “fragmentados”, e os genes não recuperados são classificados como “ausentes”, os BUSCO *missing genes* (SIMÃO *et al.*, 2015).

O método de Mann-Whitney (BIRD; BIRD, 2019) foi escolhido porque

se encaixa em situações com poucas amostras e para eventos independentes. Além disso, o teste de normalidade nos dados teve resultado negativo, reforçando a escolha por um teste não paramétrico. Foram estabelecidas duas hipóteses: H_0 (ou hipótese nula) em que geralmente afirma-se a igualdade ou, no caso, equivalência; e H_1 (ou hipótese alternativa) em que normalmente testa-se $>$, $<$ou $\neq$ para tentar reunir provas suficientes para rejeitar H_0.

Então a probabilidade de H_0 ser verdadeira foi calculada (através do *valor-p*). Estabelece-se um valor α, que é a menor probabilidade de se acreditar que a hipótese nula é verdadeira. Se $valor-p < \alpha$ rejeita-se H_0, senão se $valor-p > \alpha$, falha-se em rejeitar H_0.

A pipeline do *PASA* gera um arquivo do tipo GFF3 (formato de anotação genômica que evoluiu do GFF e mantêm semelhanças com o mesmo) contendo as alterações sugeridas na anotação e da mesma forma é possível através do utilitário *gff3_merge*, do *MAKER*, gerar um arquivo desse tipo como resultado final da pipeline do *MAKER*. Portanto tivemos duas versões de arquivos com anotações (antes e depois) dos processos *MAKER* e *PASA* e comparamos através do software *GFFCompare* que revela a sensibilidade e a precisão dos dados nos níveis das bases, dos *exons*, *introns*, *intron chain*, transcritos e *locus*. A precisão do processo foi calculada da seguinte forma:

$$Precisão = TP/(TP+FP)$$

onde TP significa “true positives” (verdadeiro positivos), ou características encontradas (bases, *exons*, *introns*, transcritos, etc.) que coincidem com as correspondentes características referenciadas na anotação; FN significa “false negatives” (falso negativos), por exemplo características que não estão presentes na entrada de dados; FP, “false positives” (falso positivos) são as características presentes na entrada de dados mas não confirmados por qualquer dado referenciado na anotação. Note que FP + TP representa o conjunto completo de características encontradas no arquivo de entrada (PERTEA; PERTEA, 2020). A listagem contendo o comando utilizado para gerar esta comparação encontra-se no Apêndice C.1.

Em relação ao genoma de *C. arabica*, foram executadas duas pipelines: uma com dados de transcriptoma obtidos no NCBI e outra com dados de transcriptoma de IVAMOTO *et al.* (2017), ambas com genoma do NCBI. Portanto, foram realizados quatro experimentos com os arquivos de genoma das três espécies de café: um para *C. eugenioides*, um para *C. canephora* e dois para *C. arabica*.

O software Quast também foi utilizado para produzir relatórios a respeito de k-mers e os dos gráficos sobre *NGx e Misassembles* (descritos a seguir), a fim de melhorar a confiabilidade dos resultados:

- NGx, Genome Nx: O tamanho de contig tal que usando contigs desse tamanho ou maiores alcança-se x% do tamanho do genoma de referência (GUREVICH *et al.*, 2013);
- FRCurve (misassemblies): porcentagem do genoma coberto por misassembled contigs (contigs que contém pontos montados incorretamente).

Neste trabalho, quando o termo "*missing genes*" é mencionado, ele significa BUSCO *missing genes* e não genes eventualmente perdidos durante o processamento das pipelines.

Os arquivos resultantes GFF3 revelaram uma lista de Stable IDs de genes (que são sua identificação), exclusivas de cada software, cuja saída do *PASA* foi introduzida no site do Ensembl Plants (EMBL-EBI, 2019) e gerou uma planilha eletrônica de recursos extraídos, como um arquivo XLS. As informações nesta tabela são:

- Gene Stable ID
- Transcript Stable ID
- GO Term Accession
- Go Term Name
- Gene Description

- Gene Start (bp)
- Gene End (bp)
- Gene % GC Content
- Source (Gene)
- Transcript Length (including UTRs and CDS)

Isso nos permitiu examinar a localização dos genes encontrados para saber se o software é capaz de encontrar genes tanto na região do centrômero quanto na região do telômero com o mesmo desembaraço, o que seria uma vantagem e indicador de eficiência.

A ferramenta *PASA* também gera gráficos ROC, que ajudaram a medir seu desempenho. A análise ROC fornece ferramentas numéricas e gráficas com suporte estatístico para caracterizar o desempenho dos preditores. A curva ROC mostra a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos, respectivamente nos eixos X e Y. Tem a ver com a precisão da predição dos genes através da pipeline: o verdadeiro positivo significa o gene previsto corretamente e e falso positivo significa os genes previstos incorretamente.

4.4 ANÁLISES COMPLEMENTARES

As técnicas usadas para analisar a qualidade de uma montagem de transcriptoma também foram usadas para anotações de genoma. Uma análise importante, sugerida por HAAS (2019b), pode ser feita usando o BLAST+ em bancos de dados importantes de proteínas como o SwissProt (ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowled- gebase/complete/uniprot_sprot.fasta.gz) e TrEMBL(ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_trembl.fasta.gz).

Foi criada uma base de dados local de proteínas e aconteceu então uma pesquisa, para cada segmento *full-length*, do alinhamento que melhor se identifica com uma proteína com a qual ocorre a homologia. Em seguida, é executada a contagem das proteínas distribuídas pelas porcentagens do tamanho de cobertura dos alinhamentos dos transcritos com o melhor *matching* na pesquisa (se uma proteína de destino corresponder a várias transcrições como seus melhores *hits*, ela é contada apenas uma vez, juntamente com a transcrição que fornece a melhor *BLAST hit score* e o maior *match length*). Todos os comandos aqui citados constam dos Apêndices C.2, C.3 e C.4.

4.5 SOLUÇÃO DE INTEGRAÇÃO E AVALIAÇÃO DE SOFTWARE DE ANOTAÇÃO GENÔMICA

Para permitir que esse experimento fosse realizado novamente, pelo menos em parte e com mais facilidade, foi criada uma solução automática para gerar informações para a comparação e verificação da eficácia de dois programas de anotação genômica.

Usando Linux, por exemplo, é necessário instalar o Bio Perl, da mesma forma os módulos:

- Venn::Chart;
- Array::Contains;
- Array::Utils;
- List::Uniq;
- Bio::Seq;
- Bio::SeqIO;

Três scripts foram escritos na linguagem *Perl*. O primeiro *script* faz um levantamento dos elementos que foram produzidos pelas pipelines ou, em outras palavras, uma lista dos genes finais. Durante esse processo, o arquivo

de anotação genômica resultante é filtrado por uma expressão regular que resulta nos nomes dos genes que são gravados em um arquivo de destino.

Devido a diferença na forma que textualmente as informações dos genes são apresentadas nos gerados gerados pelo *PASA* e *MAKER* (respectivamente GFF3 e GFF), usados como entrada para os inventários, o *script* Perl precisou ser atualizado. Cada *script* foi adaptado às características da anotação genômica de seu respectivo arquivo de entrada de dados. O arquivo GFF gerado pelo *MAKER* foi utilizado pelo segundo modelo de *script* escrito para inventário de genes. Os códigos fonte de todos os scripts aqui comentados estão nos Apêndices D.1, D.2 e D.3. Essa diferença na forma de tabular os dados, particularmente a identificação dos genes existente entre arquivos de anotação genômica GFF e GFF3 torna necessária outra configuração de expressão regular para a extração dos dados.

Finalmente, temos o programa *Ensemble Solution and Evaluation of Genomic Annotation Software* (o terceiro script). Aqui ele é usado para comparar o *PASA* e o *MAKER*, mas pode ser usado para comparar quaisquer dois programas de anotação genômica que geram arquivos GFF3 e/ou GFF.

Essa solução gera os seguintes relatórios:

- onlySW1.txt: lista dos genes anotados pela pipeline do software 1 (exclusivamente);
- onlySW2.txt: lista dos genes anotados pela pipeline do software 2 (exclusivamente);
- intersec.txt: lista dos genes anotados por ambas pipelines (intersecção do conjunto);
- reportonlySW1.txt: informações provindas do GenBank sobre os genes presentes exclusivamente na pipeline do software 1;
- reportonlySW2.txt: informações provindas do GenBank sobre os genes presentes exclusivamente na pipeline do software 2;

E dois diagramas comparativos, mostrando a distribuição de genes anotados pelo software:

- VennChart.png
- VennHistogram.png

A solução acima e um arquivo ReadMe com explicações de uso estão publicadas no GitHub em https://github.com/geraldocantelli/ensemble desde 27 de março de 2020. Todos os scripts são configuráveis, aceitam parâmetros e estão registrados sob a GNU General Public License. Para que os relatórios mais detalhados sejam gerados é preciso realizar o download de um arquivo .GBFF do site do NCBI, mais especificamente do GenBank, que descreva a anotação da espécie em questão.

Com esse recurso, os relatórios reportonlySW1 e reportonlySW2 podem listar os nomes dos genes exclusivos de cada software tabelados com sua expressão CDS. Há um diretório no repositório do GitHub com exemplos de utilização deste programa gerando todas essas informações.

5 RESULTADOS E DISCUSSÃO

No primeiro experimento, foi realizada uma comparação entre os resultados das pipelines do *PASA* e do *MAKER* para amostras de *C. canephora* (DEREEPER *et al.*, 2014), *C. eugenioides* e *C. arabica*, ambos provindos do banco de dados do NCBI, em relação ao completude do BUSCO (comparadas evolutivamente com o genoma pré-anotado).

Para o segundo experimento, foram aplicados os testes de Mann-Whitney à completude do BUSCO, ao número de ortólogos ausentes, e aos resultados de precisão do GFFCompare. O terceiro experimento consistiu-se de duas análises:

- Curvas ROC (geradas pelo *PASA*), e gráficos NGx e sobre Misassemblies (produzidos pelo software Quast);
- relatórios a respeito de *k-mers* (Quast).

Em seguida, estima-se o número de proteínas pela porcentagem de cobertura dos máximos alinhamentos, analisando esses dados para entender como as propriedades de cada pipeline influenciaram nos resultados e qual característica é mais importante para um software de anotação genômica.

5.1 EXPERIMENTO #1

Geneticamente próximas para o gênero *Coffea*, tomates (LIN *et al.*, 2005) foram usadas em todas as ferramentas de predição e avaliação deste trabalho. (TRAN *et al.*, 2018) diz que o tomate é a espécie mais próxima do

cafeeiro com dados genômicos disponíveis, entre exemplos dos programas que tiveram as ferramentas de predição e análises configurados com genoma de tomate, temos o software BUSCO e outros como o SNAP (KORF, 2004) (usado pelo *MAKER*) e AUGUSTUS (STANKE; WAACK, 2003) (usado pelo *PASA*).

Para verificar se houve diferenças no desempenho entre o *PASA* e o *MAKER*, fizemos uma análise considerando a avaliação da completude do BUSCO para as duas pipelines, uma anotação genômica em melhores condições foi observada em *C. arabica* e *C. eugenioides* mas não em *C. canephora*, para ambas pipelines. Para essa última espécie foi usada a amostra do *Coffea Genome Hub*, já para a espécie *C. arabica* foi usado o *assembly AUK PRJEB4211 v1 (BioProject: PRJEB4211)* e para *C. eugenioides*, *BioProject: PRJNA497891*, ambos do reposit ório do NCBI.

A Tabela 5.1 exibe a comparação do desempenho da ferramenta *PASA* antes de depois da execução de sua pipeline, medida pela porcentagem de completude e pelo número de genes duplicados e fragmentados analisados pelo BUSCO. É possível notar que houve uma diminuição de completude do *C. canephora* que pode ser explicado pelo fato de essa ser a amostra com a maior qualidade de anotação. Por outro lado, houve um aumento na porcentagem de completude e de genes duplicados bem como diminuição do número de genes fragmentados tanto para *C. arabica* quanto para *C. eugenioides* evidenciando melhora na anotação.

A mesma melhora pode ser vista na Tabela 5.2, também a seguir. Pode-se perceber que o aumento na porcentagem de completude e a diminuição do número de genes fragmentados foram maiores para a pipeline do *MAKER*:

Tabela 5.1: Comparação do desempenho da ferramenta *PASA* - Completude do BUSCO

Pipeline do PASA	Completude (%)			Duplicados (#)			Fragmentados (#)		
Amostras	Antes	Depois	Aumento	Antes	Depois	Aumento	Antes	Depois	Diminuição
Coffea canephora	96.3	95.5	-0.8	38	48	10	59	52	7
Coffea arabica	95.3	96.4	1.1	1258	1677	419	34	33	1
Coffea eugenioides	96.5	98.2	1.7	147	777	630	27	13	14
Média	96.0	96.8		481	834	353	40	34.6	7.4
Total			2.0			1059			22

Fonte: Pipeline do *PASA*

Tabela 5.2: Comparação do desempenho da ferramenta *PASA* - Completude do BUSCO

Pipeline do MAKER	Completude (%)			Duplicados (#)			Fragmentados (#)		
Amostras	Antes	Depois	Aumento	Antes	Depois	Aumento	Antes	Depois	Diminuição
Coffea canephora	96.3	95.5	-0.8	38	44	6	59	52	7
Coffea arabica	95.3	98.0	2.7	1258	1665	407	34	15	19
Coffea eugenioides	96.5	98.6	2.1	147	756	609	27	11	16
Média	96.0	97.3		481	821.6	340.6	40	26	14
Total			4.0			1022			42

Fonte: Pipeline do *PASA*

Observando o índice de alteração por espécie, a menor diferença é encontrada na amostra de *C. canephora*, pois esta é a melhor amostra genômica montada, comparada às amostras de *C. arabica* e *C. eugenioides*. Portanto, a qualidade da anotação genômica influenciou diretamente na variação da completude do BUSCO antes e após a execução dos pipelines.

5.2 EXPERIMENTO #2

Trabalhando com os genomas de café, as informações de genes ausentes do BUSCO, além de sua completude foram calculadas após a execução dos pipelines e o teste de Mann-Whitney foi realizado com essas informações mais a precisão calculada pelo software GFFCompare (entre os arquivos GFF antes e depois das pipelines). Todas as experiências apresentadas aqui têm uma margem de confiança de 98,48% calculada pelo software MiniTab (MINITAB, 2019), que foi utilizado para todos os cálculos estatísticos.

Com base na premissa H_0 de que o desempenho de ambos os programas é equivalente, foram realizados dois testes com hipóteses alternativas que buscaram rejeitar essa premissa ou equivalência afirmando hora que o desempenho do *PASA* era melhor que o do *MAKER* e hora que o desempenho do *MAKER* era superior ao do *PASA*. Estabeleceu-se $\alpha = 0,05$ e calcularam-se o *valor-p* de cada hipótese alternativa. A primeira experiência analisou quanto aos genes perdidos do BUSCO (Tabela 5.3):

Tabela 5.3: Análise de Mann-Whitney sobre os genes ausentes do BUSCO, hipótese: *MAKER* tem melhor desempenho que *PASA*

Categoria	BUSCO *missing genes*	
Premissa (H_O)	Performance Equivalente	
Hypótese Alternativa (H_1)	*MAKER* > *PASA*	
Método	valor-w	*valor-p*
Não ajustado para empates	15,00	0,844

Fonte: Software Minitab e pipelines

Observando-se a Tabela 5.3 vemos que o *valor-p* é 0,844 (maior que 0,05), então não se pode rejeitar a equivalência ou igualdade. Para comprovar essa equivalência, outro experimento foi realizado, invertendo-se a hipótese alternativa:

Na Tabela 5.4 vê-se que considerando H_1 como *PASA* com melhor desempenho que o *MAKER*, o *valor-p* também é maior que 0,05 (reforçando H_0), mas em comparação com o primeiro *valor-p* da última experiência, é menor que este, isso pode indicar uma vantagem do *MAKER* neste ponto sobre o *PASA*.

Tabela 5.4: Análise de Mann-Whitney sobre os genes ausentes do BUSCO, hipótese: *PASA* tem melhor desempenho que *MAKER*

Categoria	BUSCO *missing genes*	
Premissa (H_O)	Performance Equivalente	
Hypótese Alternativa (H_1)	*MAKER* < *PASA*	
Método	Valor-W	*valor-p*
Não ajustado para empates	15,00	0,235
Ajustado para empates	15,00	0,234

Fonte: Software Minitab e pipelines

Utilizando o mesmo teste estatístico, foi observada a precisão indicada pelo software GFFCompare, mais uma vez não foi possível afirmar que uma pipeline tem melhor desempenho que a outra, pois o mesmo resultado foi encontrado para ambas hipóteses alternativas (e com $valor - p > \alpha$):

$$valor - p = 0,557$$

Por fim, tentamos o teste de Mann-Whitney com o indicador Completude do BUSCO:

$$valor - p_{MAKER > PASA} = 0,386$$

$$valor - p_{PASA\ >MAKER} = 0,718$$

Neste último experimento estatístico, o *PASA* obteve melhores resultados que o *MAKER* contudo sendo ambos valores de p maiores que 0,05 fica reforçada a hipótese de equivalência. Portanto, considerando todas essas três análises, não podemos concluir que a execução de um pipeline seja melhor que a outra. Como mostrado em comparações adicionais, ambos os programas melhoraram a qualidade da anotação genômica das amostras.

5.3 EXPERIMENTO #3

O pipeline da *PASA* gerou automaticamente gráficos das curvas ROC vistas na Figura 5.1, indicando a precisão na predição genômica das amostras, comparando-se seu estado inicial e de após as atualizações feitas pela própria pipeline.

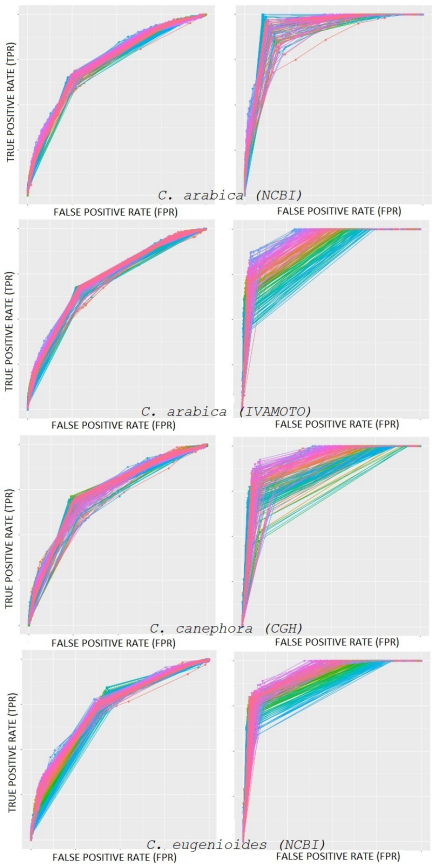

Figura 5.1: Quadro comparativo das curvas ROC para as três espécies de café entre o antes e o depois da execução da pipeline de anotação proposto neste trabalho.

Fonte: Pipeline do *PASA*

Para ter certeza de que depois da pipeline houve melhora na predição dos genes foram recuperados da ferramenta *PASA* as informações numéricas da construção da referida curva. A Tabela 5.5 mostra as médias do valor da área sob a curva ROC (AUC) antes de cada pipeline e depois das alterações promovidas por elas:

Tabela 5.5: Médias da medida AUC (*area under curve*) das curvas ROC do experimento

Amostras	Fonte	AUC (média)	
		Antes	Depois
C. arabica	Ivamoto	0,709	0,889
C. canephora	CGH	0,724	0,874
C. eugenioides	NCBI	0,711	0,892

Fonte: Pipeline do *PASA*

Quanto maior a área sob a curva ROC, maior a assertividade da predição dos genes e foi verificado isso depois da execução das pipelines. Portanto, como pode-se ver na Tabela 5.5, o software *PASA* apresentou um aumento na qualidade da anotação genômica das amostras, provavelmente devido à eficácia de seu algoritmo que trabalha com a identificação de todos os possíveis splices alternativos para gerar a anotação e resolver o problema de incerteza dos programas de predição.

Aplicando o software Quast para avaliar a anotação genômica de *C. arabica* do NCBI, antes e depois das pipelines *PASA* e *MAKER*, temos os gráficos resultantes na Figura 5.2. Na linha A, o gráfico de NGx mostra maior porcentagem do tamanho do genoma de referência com o mesmo tamanho de contig após a pipeline para o *PASA*, mas não houve alterações para o *MAKER* no depois relativo a antes da pipeline (por isso os gráficos deste software foram omitidos da referida Figura).

Sobre a porcentagem de cobertura do genoma de contigs montadas incorretamente, mostrada na linha B, para o *PASA*, a porcentagem das contigs mal montadas é menor que a do *MAKER*, que novamente não mostrou alteração em relação à situação inicial do processo. Essas duas últimas análises destacam como o *PASA* transforma especialmente a estrutura da anotação da amostra, possivelmente por causa do seu algoritmo que estipula todas as possibilidades de splices alternativos.

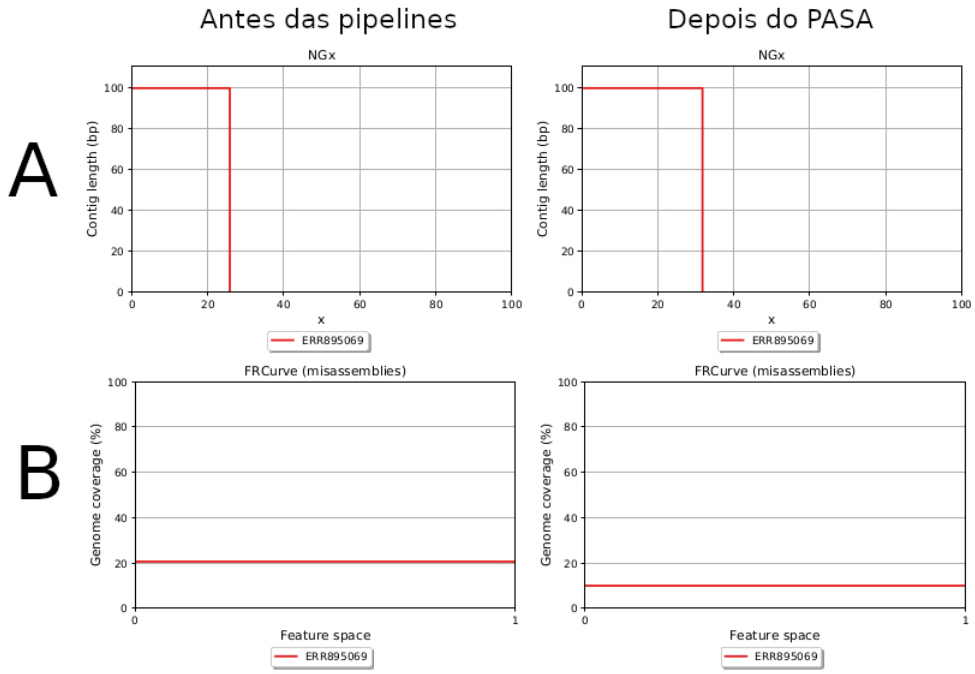

Figura 5.2: Análise do software Quast

Fonte: Software Quast

O software Quast também produziu relatórios sobre o número de *k-mers* encontrados, que pode ser visto na Tabela 5.6 (foram calculados para antes da execução das pipelines e também sobre as informações obtidas após as mesmas):

Tabela 5.6: Número de *k-mers* distintos encontrado pelo software Quast nas sequências de *C. arabica*

C. arabica	Nº total de *k-mers*	Nº de sequências
Antes	1091822262	2833
PASA	783381588	3056
MAKER	1091822262	2833

Fonte: Software Quast

A tabela 5.6 representa que o número total de *k-mers* encontrados

no genoma depois da pipeline do *PASA* é menor do que antes e que *MAKER* manteve-se constante. Na tese de doutorado de ALMEIDA (2013) foi estabelecida uma relação entre a quantidade de *k-mers* encontrada no genoma e nas sequências e o número de erros na anotação. Em seu estudo, ele apresenta que através de um genoma completamente mapeado, é possível demonstrar a relação do número de erros com o aumento da quantidade de cobertura. A Figura 5.3 apresenta um gráfico com duas linhas traçadas. A primeira, marcada com círculos, indica a quantidade de *k-mers* distintos encontrados nas bibliotecas a cada sequenciamento gerado com um certo número de cobertura. A segunda linha, marcada com losangos, mostra a quantidade de *k-mers* distintos encontrados nas bibliotecas que de fato fazem parte do genoma. A diferença entre a quantidade de *k-mers* distintos encontrados nas bibliotecas versus a quantidade de *k-mers* distintos existentes no genoma é o total de erros de sequenciamento existente.

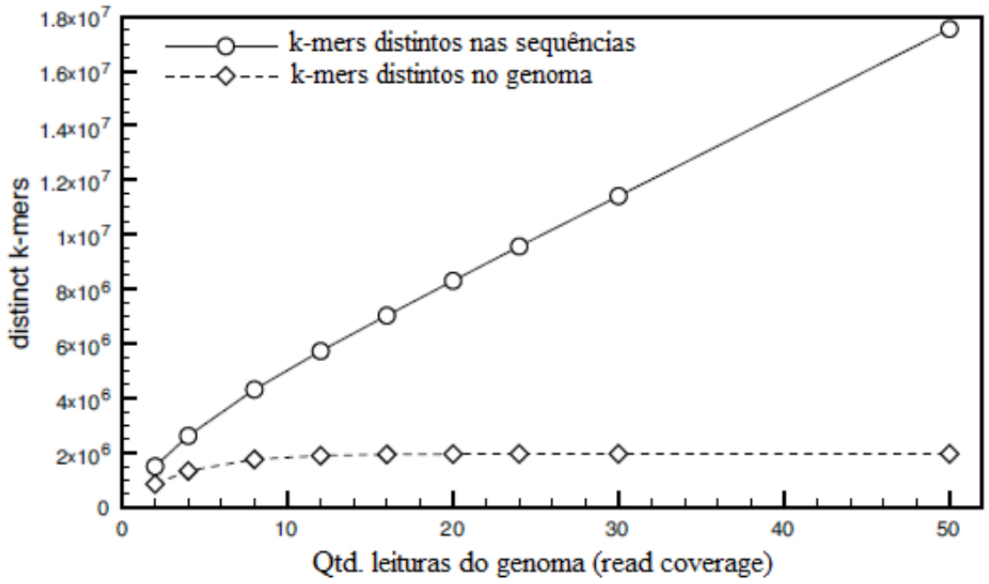

Figura 5.3: Quantidade de k-mer distintos no genoma versus quantidade de k-mer distintos sequenciados.

Fonte: (ALMEIDA, 2013)

Portanto há uma evidência de que o *PASA* diminuiu a quantidade de

erros na anotação visto que o número de *k-mers* nas sequências é menor depois de sua execução. Sobre os *k-mers* encontrados após a pipeline do *MAKER*, praticamente a mesma situação é encontrada no final, se comparada com o início do processo. A única diferença para o *MAKER* é o número de *super k-mers* (pequena diferença). Portanto, a melhoria alcançada pelo *PASA* na análise de *k-mers* não pode ser percebida nessa pipeline.

Os relatórios completos sobre *k-mers* de antes das pipelines e depois de cada uma estão nos Apêndices C.6, C.7 e C.8. Essa é mais uma evidência da eficiência do algoritmo do *PASA* que privilegia a informação de todos os possíveis splices alternativos para orientar o trabalho de anotação.

5.4 EXPERIMENTO #4

Usando o BLAST+ com o banco de dados SwissProt com o genoma de *C. arabica* antes e depois das duas pipelines, poderíamos ampliar a comparação entre o desempenho de *PASA* e *MAKER*. A Tabela 5.7 mostra evidências do experimento:

Tabela 5.7: Número de proteínas por porcentagem de cobertura nos máximos alinhamentos - *C. arabica* (Antes e Depois das pipelines)
Legenda:
%: faixa de porcentagem do alinhamento máximo coberto pelas proteínas encontradas
"bin" indica o número de proteínas nesse intervalo de porcentagem
"bin_below" indicou soma de proteínas na faixa percentual anterior.

%	Antes		Depois *PASA*		Depois *MAKER*	
	bin	bin_below	bin	bin_below	bin	bin_below
100	40	40	1373	1373	40	40
90	12	52	527	1900	12	52
80	13	65	461	2361	13	65
70	17	82	692	3053	17	82
60	24	106	835	3888	24	106
50	17	123	1227	5115	17	123
40	22	145	1608	6723	22	145
30	7	152	2169	8892	7	152
20	11	163	2580	11472	11	163
10	3	166	814	12286	3	166

Fonte: Pipelines de PASA e MAKER

Após a execução da pipeline do *PASA*, o número de proteínas com homologia encontrada é muito mais expressivo, inclusive na seção 100%. Isso prova a melhoria na qualidade da anotação genômica da amostra.

Observe que as colunas "Antes" e "Depois *MAKER*" apresentam os mesmos resultados, então concluímos que não há diferença no desempenho de antes e depois da pipeline do *MAKER* nesse quesito. Considerando os resultados do *PASA* neste tópico, concluímos que ele melhorou a qualidade da anotação genômica da amostra devido ao seu foco em trabalhar com todas os *splicing* alternativos possíveis. O *MAKER* não usa essa função dentro de seu algoritmo para corrigir a precisão da predição genômica. A anotação genômica das atualizações geradas agora pode ser disponibilizada em bancos de dados biológicos em todo o mundo.

5.5 *ENSEMBLE SOLUTION SCRIPT*

O programa “Software *Ensemble Solution* para a Avaliação de Software de Anotação Genômica” foi criado e aplicado para integrar os resultados de ambas as pipelines e gerar relatórios e diagramas que permitissem discutir as informações geradas.

A Figura 5.4 exibe um diagrama de Venn com os conjuntos de genes encontrados nas pipelines de *PASA* e *MAKER* aplicados ao genoma de *C. canephora* provindo do NCBI, indicando que 19159 genes estão presentes nas duas pipelines, 6413 genes apareceram exclusivamente na pipeline do *PASA* e nenhum apareceu exclusivamente na pipeline do *MAKER*. Resultados idênticos se refletiram no histograma de Venn que também foi gerado pelo software (vide Apêndice D.1).

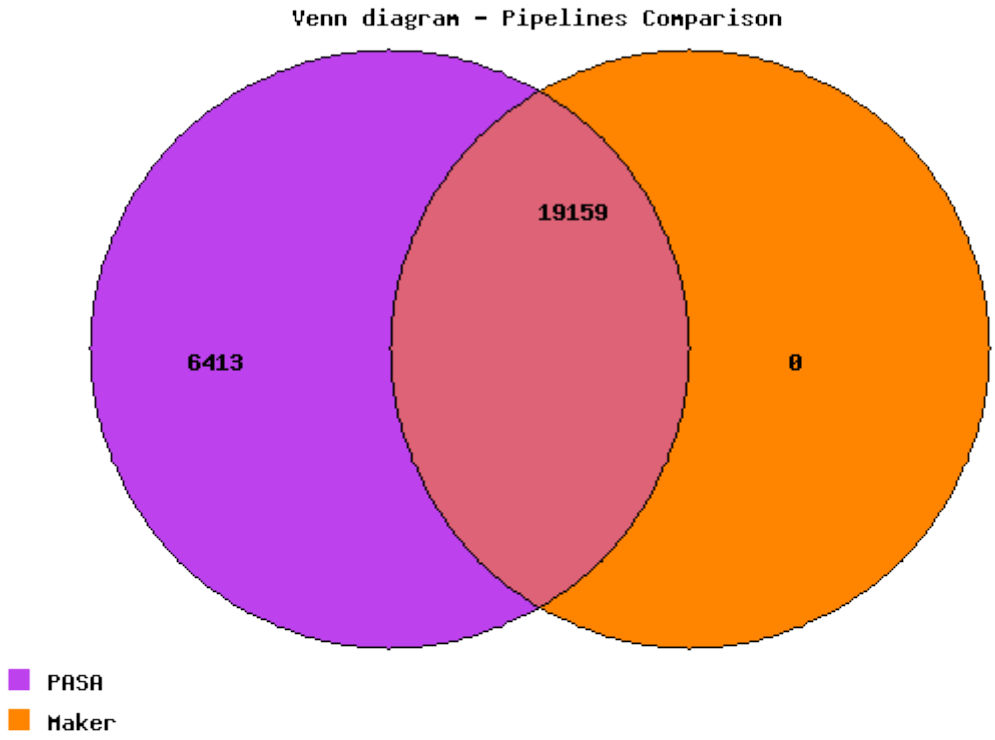

Figura 5.4: Diagrama de Venn - Número de genes encontrados nas pipelines *PASA* e *MAKER*

Fonte: Software Ensemble Solution para Comparação de Software de Anotação Genômica

Nossos resultados mostram que o *PASA* pode identificar mais genes exclusivos. É importante lembrar que o *MAKER* foi usado aqui apenas com dados de entrada genômicos (arquivos do genoma de café: FASTA e GFF3). A intenção é destacar o impacto do uso da estimativa de todos os splices alternativos pelo algoritmo *PASA*, diante da não utilização deste como método.

O software também gerou arquivos de relatório com informações sobre os genes encontrados exclusivamente por cada pipelines e sobre a intersecção do conjunto geral, ou seja, genes encontrados por ambos processos. Esse arquivo para o relatório exclusivo do *MAKER* ficou vazio.

Regiões próximas aos centrômeros normalmente têm um número mais alto de DNA repetitivo, transposons e um número menor de genes. É importante observar que, devido a trechos de DNA repetitivos, também é mais difícil ter uma boa montagem de genoma nessas regiões e, consequentemente, uma montagem inadequada de genoma também influencia a anotação. Através da informação da planilha extraída do site *Ensembl Plants* foi possível observar que a anotação do genoma gerada pelo *PASA* para *C. canephora* (dados de *start e stop codon*) se distribui igualmente por todos os locais de todos os cromossomos: tanto nos centrômeros quanto nos telômeros. Conclui-se daí que o *PASA* apresenta a mesma capacidade de localizar genes em todas as partes do cromossomo, quando muitos programas de anotação genômica não conseguem fazer isso. Essa pode ser uma vantagem adicional do *PASA* relacionada a outros programas, pertencente à sua capacidade de detectar splices alternativos como uma maneira de organizar a anotação.

6 CONSIDERAÇÕES FINAIS

Este trabalho propõe estudar qual propriedade é mais desejável em um software de anotação genômica e para isso comparou o *PASA* e o *MAKER*, ambos trabalhando em amostras de *C. canephora*, *C. eugenioides* e *C. arabica* (de mais de uma fonte de bancos de dados genômicos/transcriptômicos).

Programas que avaliam anotações genômicas como BUSCO e Quast foram utilizados para alimentar as análises estatísticas de Mann-Whitney. Valores mensuráveis (numéricos) e também gráficos foram gerados, como as curvas ROC e o gráfico NGx. Dados recuperados do site da *Ensembl Plants* permitiram investigar a distribuição da localização por parte do *PASA* dos genes em todo o cromossomo da amostra do gênero *Coffea*.

Os testes estatísticos revelaram similaridade no desempenho de ambos os programas, *PASA* e *MAKER*, através da análise dos resultados de suas pipelines. No entanto, o *PASA* é mais sensível para detectar genes em todas as regiões cromossômicas e detecta um maior número de genes exclusivos e também de proteínas devido ao seu método de estimar todos os *splicing* alternativos para seu funcionamento. Este trabalho mostra a importância dessa técnica para a sensibilidade de uma ferramenta de anotação genômica.

Uma nova ferramenta para análise de programas de anotação genômica foi desenvolvida para poder possilitar que outros programas possam ser comparados, inclusive com geração de gráficos de Venn e relatórios sobre genes exclusivos de cada pipeline, e agora está disponível na Plataforma GitHub (https://github.com/geraldocantelli/ensemble). Novas versões de arquivos FASTA do genoma de *C. arabica*, *C. eugenioides* e *C. canephora* também es-

tarão disponíveis no banco de dados de genoma do Instituto Europeu de Bioinformática (EMBL-EBI).

REFERÊNCIAS

ABASTECIMENTO, M. Agricultura Pecuária e. **Café**: Café no brasil. 2020. Disponível em: http://antigo.agricultura.gov.br/assuntos/politica-agricola/cafe.

ADAMS, M. *et al.* Complementary dna sequencing: expressed sequence tags and human genome project. **Science**, v. 252, p. 1651–1656, 1991.

ALMEIDA, A. A. M. **Novas abordagens para o problema do alinhamento múltiplo de sequências**. Tese (Doutorado) — Pontificia Universidade Catolica do Rio de Janeiro, Rio de Janeiro, 2013.

ARMSTRONG, J. *et al.* Whole-genome alignment and comparative annotation. **Annual Review of Animal Biosciences**, v. 7, n. 1, p. 41–64, 2019. ISSN 2165-8102.

BIRD, J.; BIRD, J. The Mann-Whitney test. In: **Mathematics Pocket Book for Engineers and Scientists**. [S.l.: s.n.], 2019.

BRADNAM, K. R. *et al.* Assemblathon 2: Evaluating de novo methods of genome assembly in three vertebrate species. **GigaScience**, v. 2, n. 1, p. 1–31, 2013.

CAMPBELL, M. *et al.* Automated update, revision and quality control of the zea mays genome annotations using maker-p improves the b73 refgen_v3 gene models and identifies new gene models. **Plant Physiology**, 2014.

CANTAREL, B. *et al.* Maker: an easy-to-use annotation pipeline designed for emerging model organism genomes. **Genome Res**, v. 18, p. 188–96, 2008. PubMed: 18025269.

CARMO, V. **Teste de Mann-Whitney**. 2019. Acessado em: 03/12/2019. Disponível em: http://www.inf.ufsc.br/~vera.carmo/Testes_de_Hipoteses/Testes_nao_parametricos_Mann-Whitney.pdf.

CRUZ, U. of C. S. **USCS Sequence and Annotation Downloads**. 2019. Acessado em: 28/11/2019. Disponível em: https://hgdownload.soe.ucsc.edu/downloads.html.

DARLING, A. E. *et al.* Mauve assembly metrics. **Bioinformatics**, v. 27, n. 19, p. 2756–2757, 2011.

DEREEPER, A. *et al*. The coffee genome hub: a resource for coffee genomes. **Nucleic Acids Research**, v. 43, n. D1, p. D1028–D1035, 11 2014. ISSN 0305-1048. Disponível em: https://doi.org/10.1093/nar/gku1108.

EARL, D. *et al*. Assemblathon 1: A competitive assessment of de novo short read assembly methods. **Genome Research**, v. 21, n. 12, p. 2224–2241, 2011.

EMBL-EBI. **Ensembl Plants Web Site**. 2019. Acessado em: 03/12/2019. Disponível em: https://plants.ensembl.org/info/website/index.html.

EYRAS, E. *et al*. The ensembl automatic gene annotation system. **Genome Research**, v. 14, n. 5, p. 942–950, 2004.

GUREVICH, A. *et al*. Quast: Quality assessment tool for genome assemblies. **Bioinformatics**, v. 29, n. 8, p. 1072–1075, 2013.

HAAS, B. J. **GitHub - PASA Software**. 2019. Acessado em: 29/11/2019. Disponível em: https://github.com/PASApipeline/PASApipeline.

HAAS, B. J. **GitHub trinityrnaseq/trinityrnaseq**. 2019. Acessado em: 02/03/2019. Disponível em: https://github.com/trinityrnaseq/trinityrnaseq/wiki/Counting-Full-Length-Trinity-Transcripts.

HAAS, B. J. *et al*. Improving the arabidopsis genome annotation using maximal transcript alignment assemblies. **Nucleic Acids Research**, v. 31, n. 19, p. 5654–5666, 2003.

HAAS, B. J. *et al*. Automated eukaryotic gene structure annotation using evidencemodeler and the program to assemble spliced alignments. **Genome Biology**, v. 9, n. 1, p. 1–22, 2008.

HAAS, B. J. *et al*. Approaches to fungal genome annotation. **Mycology**, v. 2, n. 3, p. 118–141, 2011.

HOLT, C.; YANDELL, M. Maker2: an annotation pipeline and genome-database management tool for second-generation genome projects. **BMC Bioinformatics**, v. 12(1):491, 2011. PubMed: 16093699.

HUNT, M. *et al*. Reapr: A universal tool for genome assembly evaluation. **Genome Biology**, v. 14, n. 5, 2013.

IVAMOTO, S. T. *et al*. Transcriptome analysis of leaves, flowers and fruits perisperm of coffea arabica l. reveals the differential expression of genes involved in raffinose biosynthesis. **PLoS ONE**, v. 12, n. 1, p. e0169595, 2017. ISSN 19326203.

JENUTH, J. P. **The NCBI. In: Misener S., Krawetz S.A. (eds) Bioinformatics Methods and Protocols. Methods in Molecular Biology™**. Totowa, NJ.: Humana Press, 2000.

KORF, I. Gene finding in novel genomes. **BMC Bioinformatics**, v. 5, n. 1, p. 59, 2004.

KULP, D. *et al.* A generalized hidden markov model for the recognition of human genes in dna. **Proc. Int. Conf. Intelligent Syst. Mol. Biol**, v. 4, p. 134–42, 1996.

LETOVSKY, S. *et al.* Gdb: The human genome database. **Nucleic Acids Res**, v. 26, p. 94–99, 1998. PubMed: 9399808.

LIN, C. *et al.* Coffee and tomato share common gene repertoires as revealed by deep sequencing of seed and cherry transcripts. **Theoretical and Applied Genetics**, v. 112, n. 1, p. 114–130, 2005.

LUKASHIN, A.; BORODOVSKY, M. Genemark.hmm: new solutions for gene finding. **Nucleic Acids Res**, v. 26, p. 1107–15, 1998. PubMed: 9461475.

MCGUIRE, A. *et al.* **Cross-kingdom patterns of alternative splicing and splice recognition**: Genome biology. 2008. R50 p. Disponível em: http://www.ncbi.nlm.nih.gov/pubmed/18321378.

MINITAB, L. **Interpretar os principais resultados para Teste de Mann-Whitney - Suporte ao Minitab 19**. 2019. Acessado em: 03/12/2019. Disponível em: https://support.minitab.com/pt-br/minitab/19/help-and-how-to/statistics/nonparametrics/how-to/mann-whitney-test/interpret-the-results/key-results/?SID=115100.

MORENO, P. A. *et al.* Bioinformatics software for genomic: a 1 systematic review on github. **Systems and Computer Engineering School, Faculty of Engineering, Universidad del Valle, Cali, Colombia**, 2018.

NARZISI, G.; MISHRA, B. Comparing de novo genome assembly: The long and short of it. **PLoS ONE**, v. 6, n. 4, 2011.

NCA, N. C. A. U. **The Economy Impact of The Coffee Industry**. 2019. Acessado em: 13/06/2019. Disponível em: http://www.ncausa.org/Industry-Resources/Economic-Impact.

NEUMANN, G. B. **A Framework Approach for Quality Feature Analysis of Genome Assemblies**. 79 p. Dissertação (Mestrado) — Pontificia Universidade Catolica do Rio de Janeiro, Rio de Janeiro, 2019.

PEARSON, W. R. Finding protein and nucleotide similarities with fasta. **Current Protocols in Bioinformatics**, v. 53, p. 1–25, 2016.

PEARSON, W. R. **Git repository for FASTA36 sequence comparison software**. 2019. Acessado em: 03/12/2019. Disponível em: https://github.com/wrpearson/fasta36.

PEARSON, W. R.; LIPMAN, D. J. Rapid and sensitive protein similarity searches. **Science**, v. 227, n. 4693, p. 1435–1441, 1985.

PERTEA, G.; PERTEA, M. Gff utilities: Gffread and gffcompare. **F1000Research**, v. 9, p. 304, 2020.

PHILLIPPY, A. M.; SCHATZ, M. C.; POP, M. Genome assembly forensics: Finding the elusive mis-assembly. **Genome Biology**, v. 9, n. 3, 2008.

PRUITT, K.; TATUSOVA, T.; MAGLOTT, D. Ncbi reference sequences (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins. **Nucleic Acids Res**, v. 35, p. D61–D65, 2006. PubMed: 17130148.

QUACKENBUSH, J. *et al.* The tigr gene indices: analysis of gene transcript sequences in highly sampled eukaryotic species. v. 29, p. 159–164, 2001.

RAHMAN, A.; PACHTER, L. Cgal: Computing genome assembly likelihoods. **Genome Biology**, v. 14, n. 1, p. R8, 2013.

SALZBERG, S. L. Next-generation genome annotation: We still struggle to get it right. **Genome Biology**, Genome Biology, v. 20, n. 1, p. 19–21, 2019.

SEKI, M. *et al.* Improving the arabidopsis genome annotation using maximal transcript alignment assemblies. **Science**, v. 296, n. 5565, p. 141–5, 2002.

SIMÃO, F. A. *et al.* Busco: assessing genome assembly and annotation completeness with single-copy orthologs. **Bioinformatics**, v. 31, n. 19, p. 3210–3212, 06 2015. ISSN 1367-4803. Disponível em: https://dx.doi.org/10.1093/bioinformatics/btv351.

SLATER, G. S. C.; BIRNEY, E. Automated generation of heuristics for biological sequence comparison. **BMC Bioinformatics**, v. 6, p. 1–11, 2005.

STANKE, M.; WAACK, S. Gene prediction with a hidden Markov model and a new intron submodel . **Bioinformatics**, v. 19, n. suppl_2, p. ii215–ii225, 09 2003. ISSN 1367-4803. Disponível em: https://dx.doi.org/10.1093/bioinformatics/btg1080.

TRAN, H. T. *et al.* Use of a draft genome of coffee (coffea arabica) to identify snps associated with caffeine content. **Plant Biotechnology Journal**, v. 16, n. 10, p. 1756–1766, 2018.

VIEIRA, L. G. E. *et al.* Brazilian coffee genome project: an est-based genomic resource. **Brazilian Journal of Plant Physiology**, v. 18, n. 1, p. 95–108, 2006.

WU, T. **GMAP and GSNAP**. 2019. Acessado em: 03/12/2019. Disponível em: http://research-pub.gene.com/gmap/.

YANDELL, M.; ENCE, D. A beginner's guide to eukaryotic genome annotation. **Nature Reviews Genetics**, v. 13, n. 5, p. 329–342, 2012. ISSN 14710056. Disponível em: http://dx.doi.org/10.1038/nrg3174.

YANDELL, M. *et al*. Large-scale trends in the evolution of gene structures within 11 animal genomes. **PLoS Comput. Biol.**, 2006.

APÊNDICE A – PIPELINE DE MONTAGEM DE ALINHAMENTOS E ANOTAÇÃO DO *PASA*

A sequência de entrada de transcritos foi selecionada e aparada para regiões de sequência de baixa qualidade e caudas poli (A) usando os protocolos de limpeza de sequência TIGR Gene Indices (QUACKENBUSH *et al.*, 2001) implementada na ferramenta *SeqClean*.

Depois de removidas as regiões de sequências irrelevantes de transcritos, as sequências de entrada foram alinhadas contra o genoma completo de cada uma das três amostras de café usando programas de alinhamento de cDNA: BLAT e GMAP, como indica o esquema da Figura A.1:

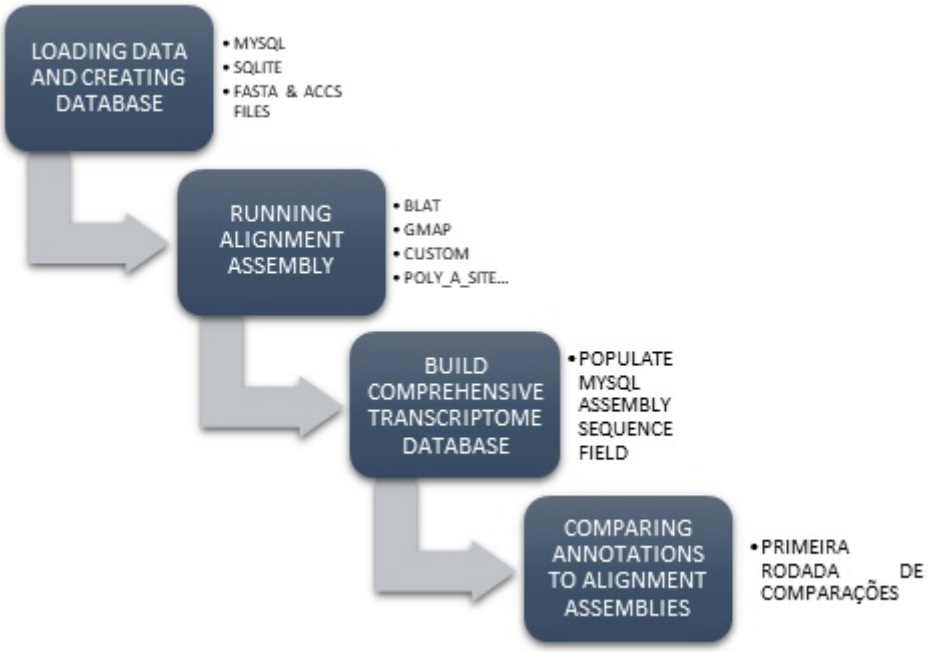

Figura A.1: Pipeline Principal *PASA* - 1ª parte

Fonte: O Autor

O comando que inicia a pipeline é o seguinte (é preciso também certificar-se que os arquivos de genoma foram renomeados para genome_sample.fasta, os arquivos de transcriptoma renomeados para all_transcripts.fasta, e os arquivos GFF3 para orig_annotations_sample.gff3 antes de iniciarem-se os processos):

Listing A.1: Comando inicial da pipeline do *PASA*

```
1 ../Launch_PASA_pipeline.pl -c $align_assembly_config_file -C -r -R
    -g genome_sample.fasta -t all_transcripts.fasta.clean -T -u
    all_transcripts.fasta -f FL_accs.txt --ALIGNERS $ALIGNERS --CPU
    $CPU -N $num_top_hits --TDN tdn.accs  --
    IMPORT_CUSTOM_ALIGNMENTS_GFF3 custom_alignments.gff3
```

$CPU configura o número de processos concorrentes que serão iniciados em paralelo para cada passo importante da pipeline.

Em $num_top_hits estará o número de alinhamentos de *splicing* alternativo com top score (default: 2)

A opção –just_align_assembly configura que o sistema funcione apenas até atingir sua parte inicial de montagem de alinhamento.

Iniciado o processo, serão executados os seguintes comandos mostrados na Figura A.2:

Figura A.2: Tela Inicial de Execução do *PASA*

Fonte: O Autor

A primeira linha indica o começo do *Running Alignment Assembly*, seguido da instrução da Figura A.2. Então ocorre a conexão com o banco de dados no diretório /tmp com o nome indicado no arquivo de configuração alignAssembly.config.

De dentro do diretório *scripts*, é invocado o procedimento *create_sqlite_cdnaassembly_db.dbi* que cria os objetos do banco de dados e em seguida *upload_transcript_data.dbi* faz o carregamento dos arquivos de entrada para o banco.

Só então é dado o comando para a chamada dos *aligners* blat e gmap, através do *script* em Perl chamado *run_spliced_aligners.pl* e depois o processo do software blat é iniciado.

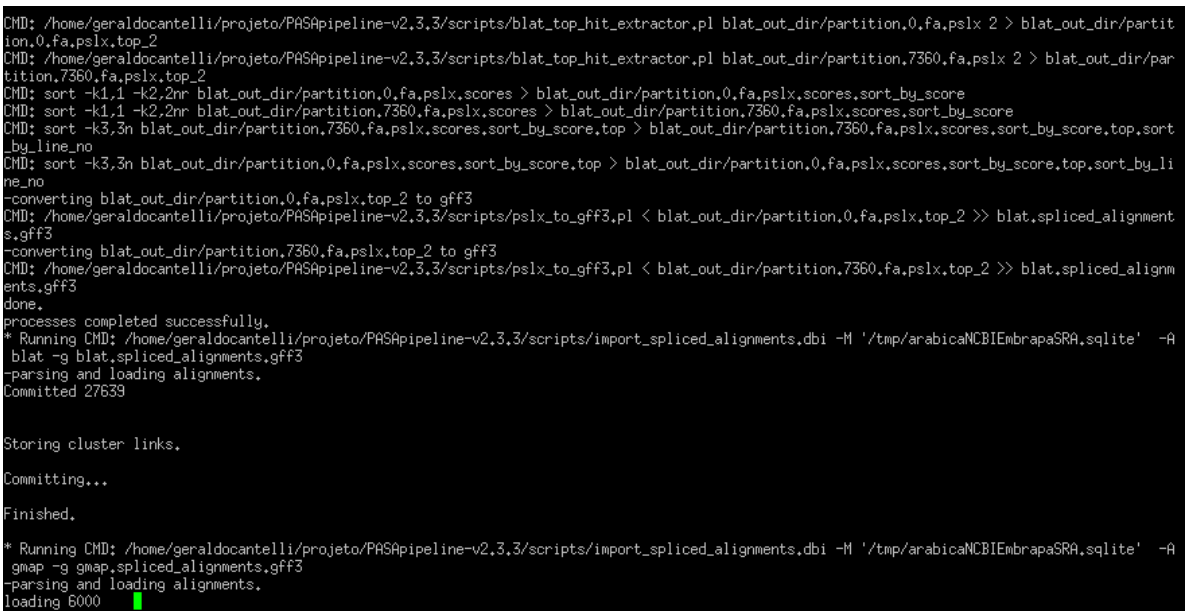

Figura A.3: Aligners blat e gmap em ação

Fonte: O Autor

Já na Figura A.3 tem-se a continuação do processamento de blat, notadamente com a execução do *script blat_top_hit_extractor.pl* e a criação do diretório blat_out_dir. Logo após o componente import_spliced_alignment.dbi aciona ambos aligners blat e gmap utilizando como material de trabalho o próprio banco de dados em */tmp/arabicaNCBIEmbrapaSRA.sqlite* e gerando respectivamente os arquivos:

- blat.spliced_alignments.gff3
- gmap.spliced_alignments.gff3

Figura A.4: Extração de frequências de base

Fonte: O Autor

Então o mesmo componente import_spliced_alignment.dbi passa a tratar os alinhamentos customizados, desde que essa opção tenha sido declarada no lançamento da pipeline. Neste momento é ativado o plugin Transdecoder passando a analisar o arquivo de transcritos que foi limpo por SeqClean e é dado início ao cálculo das frequências das bases.

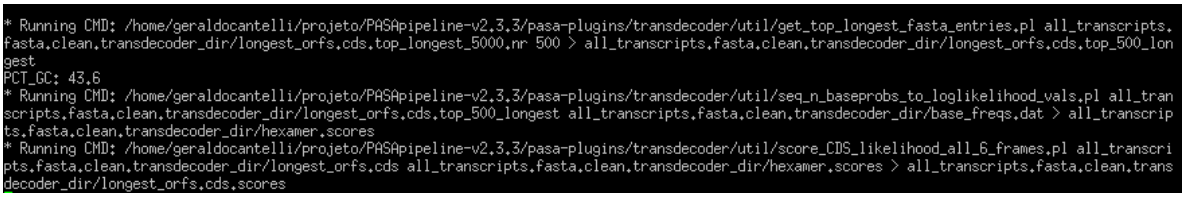

Figura A.5: CDS das 500 maiores ORFs

Fonte: O Autor

Ocorre a criação da pasta *all_transcripts.fasta.clean.transdecoder_dir* e arquivos de scores se estabelecem aí dentro. Destaque para o arquivo que contém as 500 maiores ORFs em formato CDS e o início do cálculo de métricas gravadas em arquivos chamados *scores*.

Figura A.6: Scripts para geração de estatísticas

Fonte: O Autor

Na Figura A.6, os scripts *feature_scores_to_ROC.pl* e *compute_AUC.pl* fornecem dados para que RScript possa ser invocado e criar gráficos de curvas ROC em arquivos de formato PDF.

Nota-se a importância da criação do arquivo: *all_transcripts.fasta.clean.transdecoder.gff3* a partir do arquivo de starts revisados dos melhores candidatos dos CDS das maiores ORFs.

Em seguida, o *script gff3_file_to_proteins.pl* inicia processos nesta e na próxima figura que transformam dados deste gff3 para sua versão em .bed e .pep (proteínas).

Figura A.7: Extração de Full lenght Accessions

Fonte: O Autor

A Figura A.7 representa uma continuação da anterior com o acréscimo da criação do arquivo .cds e da geração dos dados dos códigos de *Full Length Acession* (.fl_accs), também a partir do importante *all_transcripts.fasta.clean.transdecoder.gff3*.

Na seguinte Figura A.8 tem-se a ação do componente *PASA_transcripts_and_assemblies_ to_GFF3.dbi* criando arquivos não apenas no formato .gff3 mas também .bed e .gtf assim como:

- valid_blat_alignments e failed_blat_alignments
- valid_gmap_alignments e failed_gmap_alignments
- valid_custom_alignments e failed_custom_alignments

Figura A.8: Componente *PASA_transcripts_and_assemblies_ to_GFF3.dbi*

Fonte: O Autor

E finalizando esta parte da pipeline, iniciada pelo comando mostrado no Listing de código-fonte A.1, tem-se a execução de componentes que realizam o carregamento dos *assemblies* para o banco de dados, a criação e a carga dos *subclusters*. Destacam-se os componentes *alignment_assembly_to_gene _models.dbi* e *PASA_transcripts_and_assemblies_to_GFF3.dbi* que por sua vez gerará os arquivos:

- *$DBname.pasa_assemblies.gff3*
- *$DBname.pasa_assemblies.bed*
- *$DBname.pasa_assemblies.gtf*

O último arquivo gerado nesta fase é *$DBname.pasa_assemblies_described.txt* cuja importância é descrever para cada assembly, suas partes constituintes

(cDNAS, ESTs, por exemplo), sua orientação e faixa(s) em que ocorre(m) o(s) alinhamento(s).

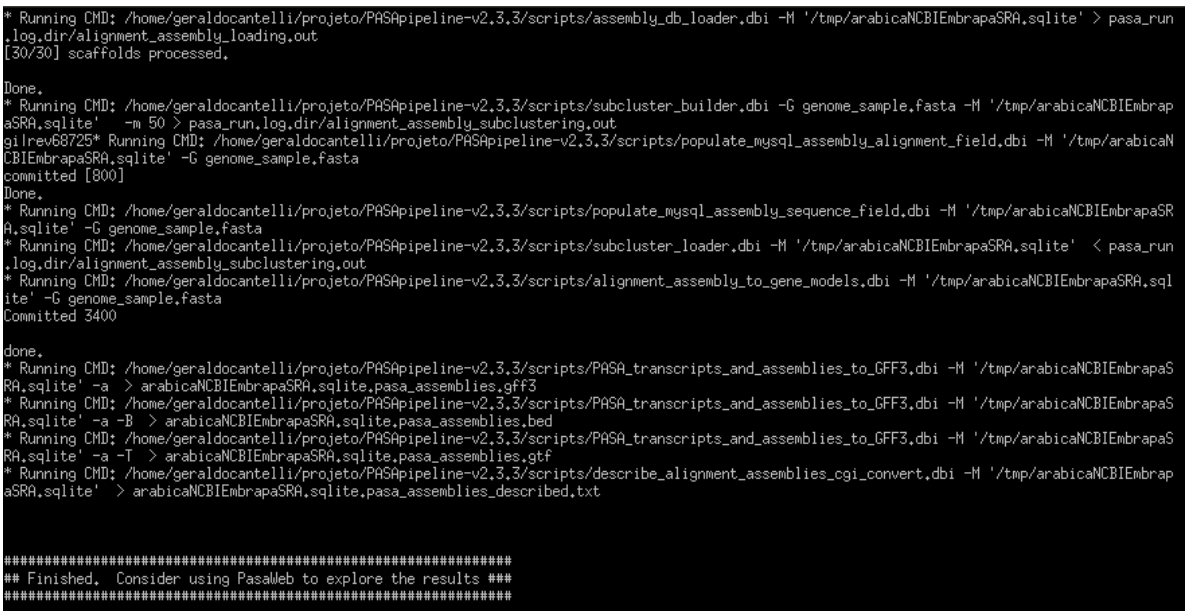

Figura A.9: Finalização da primeira parte da pipeline

Fonte: O Autor

Então inicia-se nova fase, a das comparações de anotações; a primeira é uma comparação com as anotações originais (indicadas pelo arquivo no parâmetro *annots*) :

,

Listing A.2: Comando da primeira rodada de comparações

```
1 ../Launch_PASA_pipeline.pl -c $annot_compare_config_file -g
2 genome_sample.fasta -t all_transcripts.fasta.clean -A -L --
3 annots orig_annotations_sample.gff3 --CPU $CPU
```

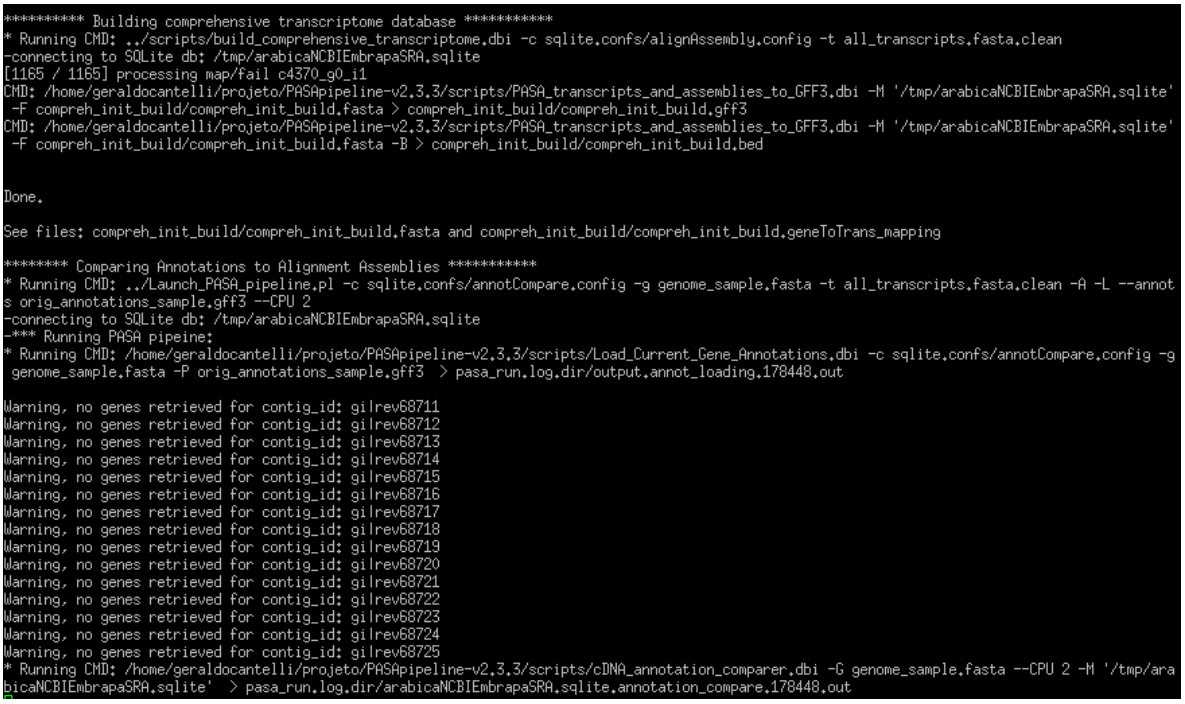

Figura A.10: Primeira rodada de comparações de anotações

Fonte: O Autor

Então iniciar-se-á a segunda parte, representada no esquema abaixo:

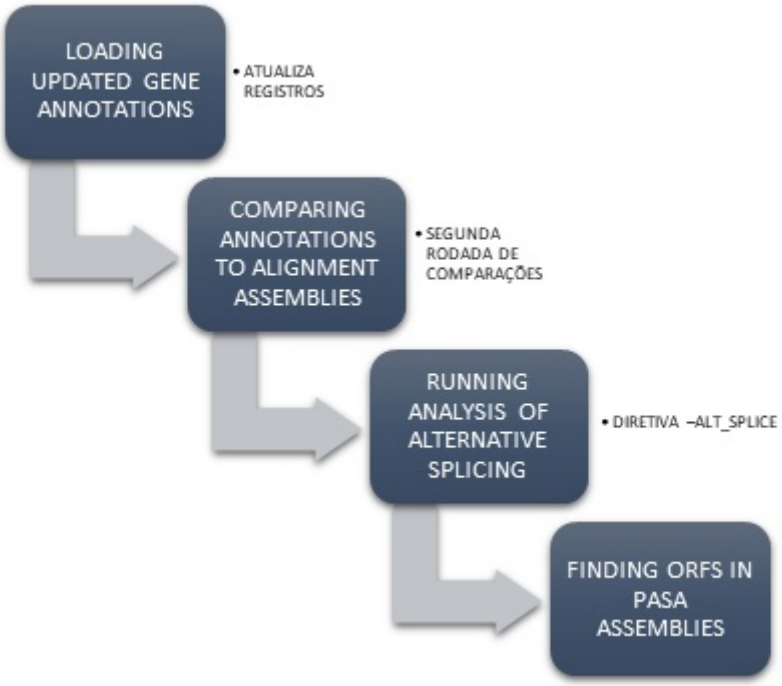

Figura A.11: Pipeline Principal *PASA* - 2ª parte

Fonte: O Autor

Durante este processo pode ocorrer a criação do seguinte arquivo:

- *$DBname.gene_structures_post_PASA_updates.number.gff3*

Este arquivo ficará zerado apenas se nenhuma estrutura de gene for atualizada pelo algoritmo do *PASA*. Operações como novos genes, *split*, *merge*, *add e extends* são as mais comuns.

Caso haja alterações, uma segunda rodada de comparações passa a ocorrer com o seguinte comando:

Listing A.3: Comando da segunda rodada de comparações

```
../Launch_PASA_pipeline.pl -c $annot_compare_config_file -g
genome_sample.fasta -t all_transcripts.fasta.clean -A -L --
annots $recent_update_file --CPU $CPU
```

Substituindo-se $recent_update_file pelo arquivo último citado. E o resultado será:

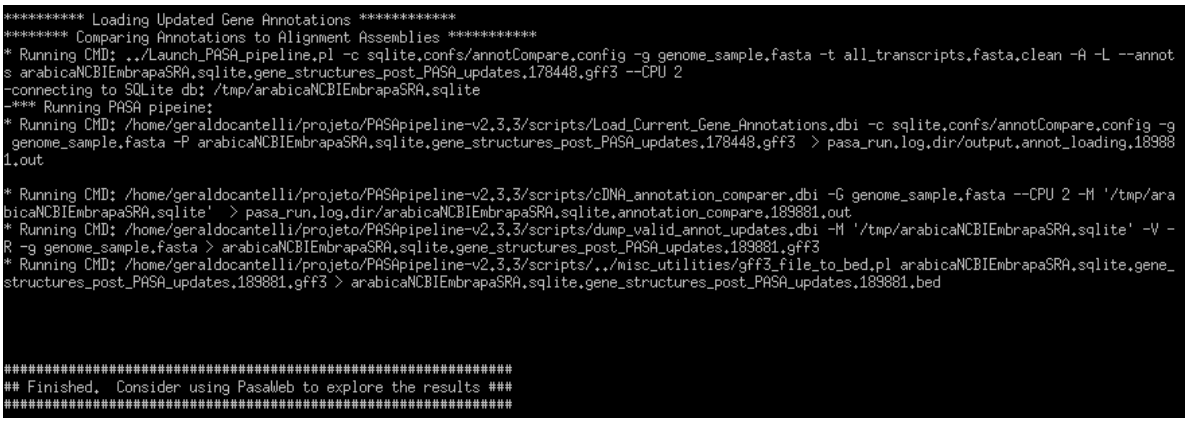

Figura A.12: Segundo rodada de comparações de anotações

Fonte: O Autor

Para proceder à busca dos resultados de splicing alternativo deve-se recorrer a seguinte instrução:

,

Listing A.4: Comando por Splicing Alternativo

```
1 ../Launch_PASA_pipeline.pl -c $annot_compare_config_file -g
2 genome_sample.fasta -t all_transcripts.fasta.clean --CPU $CPU --
3 ALT_SPLICE
```

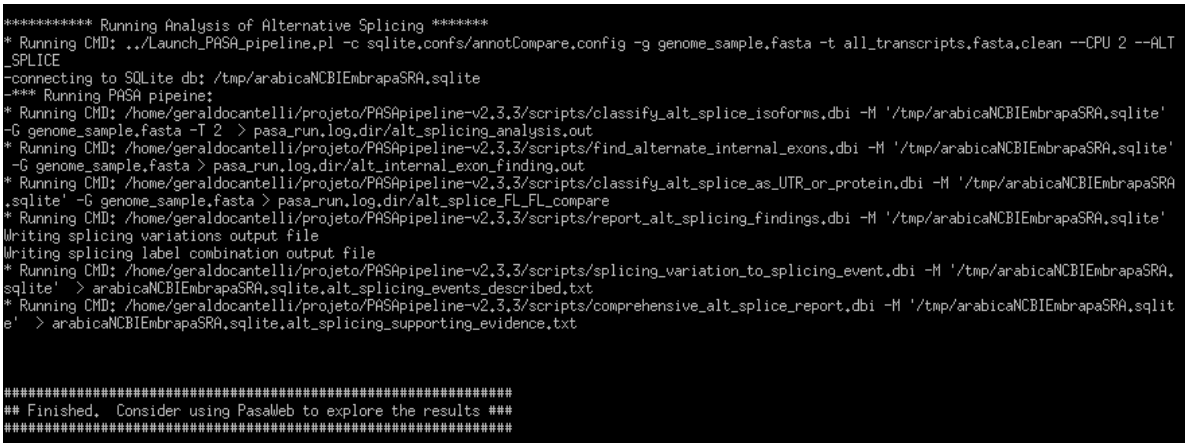

Figura A.13: Busca de resultados por Splicing Alternativo

Fonte: O Autor

A última etapa se baseia em localizar as *Open Reading Frames* e geração de significativas estatísticas para o entendimento de todo o processo e começa desta maneira:

,

Listing A.5: Comando para busca de ORFs e estatísticas

```
1 ../scripts/pasa_asmbls_to_training_set.dbi --
      pasa_transcripts_fasta
2 $DBname.assemblies.fasta --pasa_transcripts_gff3 $DBname.
3 pasa_assemblies.gff3
```

Figura A.14: Buscando ORFs: calculando frequências de bases

Fonte: O Autor

Figura A.15: Buscando ORFs: Gerando estatísticas - A

Fonte: O Autor

Figura A.16: Buscando ORFs: Gerando estatísticas - B

Fonte: O Autor

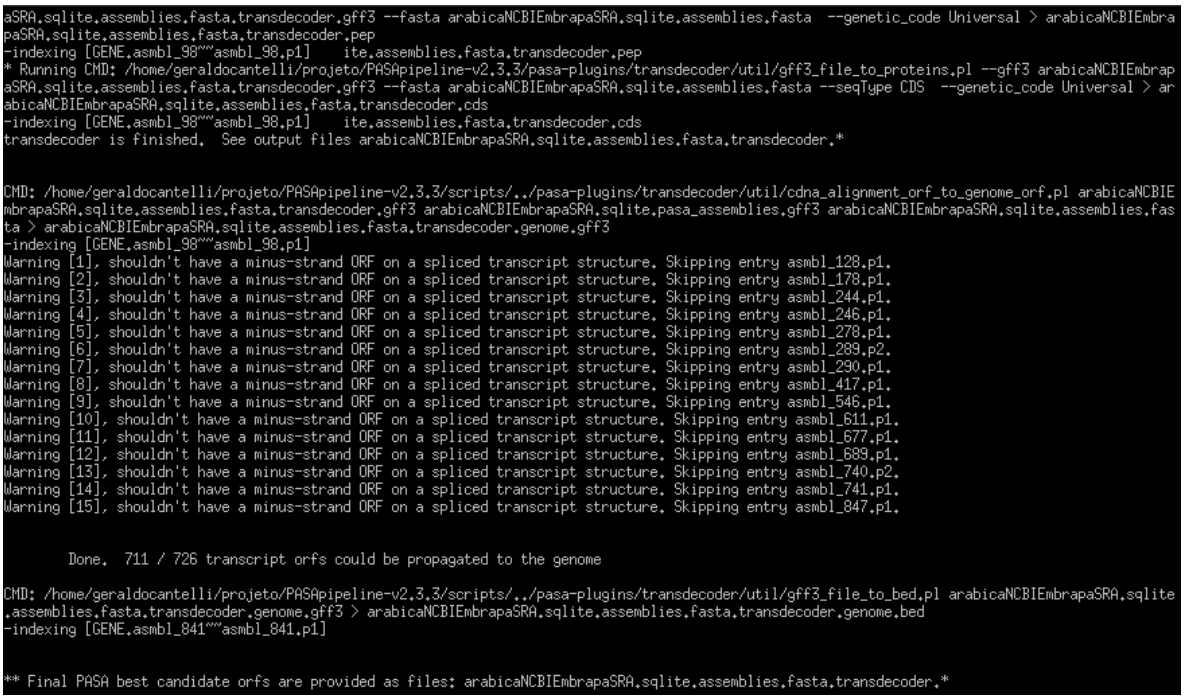

Figura A.17: Buscando ORFs: Gerando estatísticas - C

Fonte: O Autor

APÊNDICE B – PARÂMETROS DE UTILIZAÇÃO DO SOFTWARE *MAKER*

Para iniciar a pipeline, nós precisamos configurar os parâmetros:

Listing B.1: Esse comando cria três arquivos de configuração: maker_bopts.cpl, maker_exe.cpl e maker_opts.cpl

```
1 %  maker -CTL
```

Entre os arquivos criados, o único customizado para este trabalho foi maker_opts.cpl (cujas configurações estão a seguir):

Listing B.2: Conteúdo do arquivo de configuração: maker_bopts.cpl, utilizado para este trabalho

```
1  #-----Genome (these are always required)
2  genome=./genome_sample.fasta #genome sequence (fasta file or fasta
       embeded in GFF3 file)
3  organism_type=eukaryotic #eukaryotic or prokaryotic. Default is
       eukaryotic
4
5  #-----Re-annotation Using MAKER Derived GFF3
6  maker_gff= #maker derived GFF3 file
7  est_pass=0 #use ESTs in maker_gff: 1 = yes, 0 = no
8  altest_pass=0 #use alternate organism ESTs in maker_gff: 1 = yes, 0
       = no
9  protein_pass=0 #use protein alignments in maker_gff: 1 = yes, 0 =
       no
10 rm_pass=0 #use repeats in maker_gff: 1 = yes, 0 = no
11 model_pass=0 #use gene models in maker_gff: 1 = yes, 0 = no
12 pred_pass=0 #use ab-initio predictions in maker_gff: 1 = yes, 0 =
       no
```

```
13 other_pass=0 #passthrough anyything else in maker_gff: 1 = yes, 0 =
       no
14
15 #-----EST Evidence (for best results provide a file for at least
       one)
16 est= #set of ESTs or assembled mRNA-seq in fasta format
17 altest= #EST/cDNA sequence file in fasta format from an alternate
       organism
18 est_gff= #aligned ESTs or mRNA-seq from an external GFF3 file
19 altest_gff= #aligned ESTs from a closly relate species in GFF3
       format
20
21 #-----Protein Homology Evidence (for best results provide a file
       for at least one)
22 protein=  #protein sequence file in fasta format (i.e. from mutiple
       oransisms)
23 protein_gff=  #aligned protein homology evidence from an external
       GFF3 file
24
25 #-----Repeat Masking (leave values blank to skip repeat masking)
26 model_org= #select a model organism for RepBase masking in
       RepeatMasker
27 rmlib= #provide an organism specific repeat library in fasta format
       for RepeatMasker
28 repeat_protein= #provide a fasta file of transposable element
       proteins for RepeatRunner
29 rm_gff= #pre-identified repeat elements from an external GFF3 file
30 prok_rm=0 #forces MAKER to repeatmask prokaryotes (no reason to
       change this), 1 = yes, 0 = no
31 softmask=1 #use soft-masking rather than hard-masking in BLAST (i.e
       . seg and dust filtering)
32
33 #-----Gene Prediction
34 snaphmm= #SNAP HMM file
35 gmhmm= #GeneMark HMM file
36 augustus_species= tomato #Augustus gene prediction species model
37 fgenesh_par_file= #FGENESH parameter file
38 pred_gff= #ab-initio predictions from an external GFF3 file
39 model_gff= ./orig_annotations_sample.gff3 #annotated gene models
```

```
    from an external GFF3 file (annotation pass-through)
est2genome=0 #infer gene predictions directly from ESTs, 1 = yes, 0 = no
protein2genome=0 #infer predictions from protein homology, 1 = yes, 0 = no
trna=0 #find tRNAs with tRNAscan, 1 = yes, 0 = no
snoscan_rrna= #rRNA file to have Snoscan find snoRNAs
unmask=0 #also run ab-initio prediction programs on unmasked sequence, 1 = yes, 0 = no

#-----Other Annotation Feature Types (features MAKER does not recognize)
other_gff= #extra features to pass-through to final MAKER generated GFF3 file

#-----External Application Behavior Options
alt_peptide=C #amino acid used to replace non-standard amino acids in BLAST databases
cpus=1 #max number of cpus to use in BLAST and RepeatMasker (not for MPI, leave 1 when using MPI)

#-----MAKER Behavior Options
max_dna_len=100000 #length for dividing up contigs into chunks (increases/decreases memory usage)
min_contig=1 #skip genome contigs below this length (under 10kb are often useless)

pred_flank=200 #flank for extending evidence clusters sent to gene predictors
pred_stats=0 #report AED and QI statistics for all predictions as well as models
AED_threshold=1 #Maximum Annotation Edit Distance allowed (bound by 0 and 1)
min_protein=0 #require at least this many amino acids in predicted proteins
alt_splice=0 #Take extra steps to try and find alternative splicing, 1 = yes, 0 = no
always_complete=0 #extra steps to force start and stop codons, 1 = yes, 0 = no
```

```
63 map_forward=0 #map names and attributes forward from old GFF3 genes
       , 1 = yes, 0 = no
64 keep_preds=0 #Concordance threshold to add unsupported gene
       prediction (bound by 0 and 1)
65
66 split_hit=10000 #length for the splitting of hits (expected max
       intron size for evidence alignments)
67 single_exon=0 #consider single exon EST evidence when generating
       annotations, 1 = yes, 0 = no
68 single_length=250 #min length required for single exon ESTs if '
       single_exon is enabled'
69 correct_est_fusion=0 #limits use of ESTs in annotation to avoid
       fusion genes
70
71 tries=2 #number of times to try a contig if there is a failure for
       some reason
72 clean_try=0 #remove all data from previous run before retrying, 1 =
        yes, 0 = no
73 clean_up=0 #removes theVoid directory with individual analysis
       files, 1 = yes, 0 = no
74 TMP= #specify a directory other than the system default temporary
       directory for temporary files
```

Finalmente para dar início ao processo:

Listing B.3: Este comando dispara a pipeline do MAKER

```
1 %  maker
```

APÊNDICE C – COMANDOS GERAIS

Listing C.1: Gerando a comparação entre o antes e depois com GFFCompare

```
$ gffcompare -R -r  arabicaNCBI.assemblies.transdecoder.genome.gff3
-o    GFFComparePasa_arabicaNCBI orig_annotations_sample.gff3

$ gffcompare -R -r  arabicaNCBI.gff3Merge.gff3
-o    GFFCompareMaker_arabicaNCBI orig_annotations_sample.gff3
```

Listing C.2: Este comando cria os arquivos de banco de dados local UniProt, preparando o ambiente para a experiência

```
%  makeblastdb -in uniprot_sprot.fasta -dbtype prot
```

Listing C.3: Rodando a busca pelos máximos alinhamentos e gerando o arquivo blastx.outfmt6

```
% blastx -query genome_sample.fasta -db uniprot_sprot.fasta
-out blastx.outfmt6 -evalue 1e-20 -num_threads 6
-max_target_seqs 1 -outfmt 6
```

Listing C.4: Processamento do arquivo blastx.outfmt6: contabilizando a contagem das proteínas por cobertura de alinhamento e gerando relatórios

```
% $TRINITY_HOME/util/analyze_blastPlus_topHit_coverage.pl blastx.
   outfmt6 genome_sample.fasta uniprot_sprot.fasta
```

Listing C.5: Criação do arquivo GTF pelo *MAKER* a partir do comando gff3_merge

```
/opt/bioinformatics/share/maker2/maker/bin/gff3_merge -d
   genome_sample.maker.output/genome_sample_master_datastore_index.
   log
```

Listing C.6: Resultado dos k-mers antes das pipelines

```
    1st stage: 4.7829s
    2nd stage: 128.74s
    Total    : 133.523s
    Tmp size : 862MB

    Stats:
       No. of k-mers below min. threshold :           0
       No. of k-mers above max. threshold :   129927993
       No. of unique k-mers               :   816660752
       No. of unique counted k-mers       :   686732759
       Total no. of k-mers                :  1091822262
       Total no. of sequences             :        2833
       Total no. of super-k-mers          :    22269819
    1st stage: 1.73841s
    2nd stage: 0.175773s
    Total    : 1.91418s
    Tmp size : 0MB

    Stats:
       No. of k-mers below min. threshold :           0
       No. of k-mers above max. threshold :           0
       No. of unique k-mers               :           0
       No. of unique counted k-mers       :           0
       Total no. of k-mers                :           0
       Total no. of sequences             :     2824987
       Total no. of super-k-mers          :           0
```

Listing C.7: Resultado dos k-mers depois da pipeline do *PASA*

```
1     1st stage: 4.33928s
2     2nd stage: 7.3649s
3     Total    : 11.7042s
4     Tmp size : 638MB
5
6     Stats:
7        No. of k-mers below min. threshold :           0
8        No. of k-mers above max. threshold :   162468062
9        No. of unique k-mers               :   162471118
10       No. of unique counted k-mers       :        3056
11       Total no. of k-mers                :   783381588
12       Total no. of sequences             :     1165653
13       Total no. of super-k-mers          :    16726948
14    1st stage: 2.18952s
15    2nd stage: 0.124125s
16    Total    : 2.31365s
17    Tmp size : 0MB
18
19    Stats:
20       No. of k-mers below min. threshold :           0
21       No. of k-mers above max. threshold :           0
22       No. of unique k-mers               :           0
23       No. of unique counted k-mers       :           0
24       Total no. of k-mers                :           0
25       Total no. of sequences             :     2824987
26       Total no. of super-k-mers          :           0
```

Listing C.8: Resultado dos k-mers depois da pipeline do MAKER

```
1     1st stage: 4.7829s
2     2nd stage: 128.74s
3     Total    : 133.523s
4     Tmp size : 862MB
5
6     Stats:
7        No. of k-mers below min. threshold :           0
8        No. of k-mers above max. threshold :   129927993
9        No. of unique k-mers               :   816660752
10       No. of unique counted k-mers       :   686732759
```

```
11        Total no. of k-mers                :   1091822262
12        Total no. of sequences             :         2833
13        Total no. of super-k-mers          :     22269819
14     1st stage: 1.73841s
15     2nd stage: 0.175773s
16     Total    : 1.91418s
17     Tmp size : 0MB
18
19     Stats:
20        No. of k-mers below min. threshold :            0
21        No. of k-mers above max. threshold :            0
22        No. of unique k-mers               :            0
23        No. of unique counted k-mers       :            0
24        Total no. of k-mers                :            0
25        Total no. of sequences             :      2824987
26        Total no. of super-k-mers          :            0
```

APÊNDICE D – ENSEMBLE SOLUTION AND EVALUATION OF GENOMIC ANNOTATION SOFTWARE

Listing D.1: *Script* em Perl do Inventório usado para a pipeline do *PASA*

```
1  #!/usr/bin/perl
2
3  my $fonteDestino = $ARGV[0];
4  my $gravaDestino = $ARGV[1];
5  my $line;
6  open (IN, $fonteDestino) or die "cannot open file";
7  open(my $fh, '>', $gravaDestino) or die "cannot open file";
8  while ($line = <IN>){
9
10 chomp($line);
11 if ($line=~/.*gene.*Name=(.*)/){
12    print ($fh "$1\n");
13 }
14 }
15 close ($fh);
16 close (IN);
17 exit 0;
```

Listing D.2: *Script* em Perl do Inventório usado para a pipeline do MAKER

```
1  #!/usr/bin/perl
2
3  my $fonteDestino = $ARGV[0];
4  my $gravaDestino = $ARGV[1];
5  my $line;
6  open (IN, $fonteDestino) or die "cannot open file";
```

```
7  open(my $fh, '>', $gravaDestino) or die "cannot open file";
8  while ($line = <IN>){
9
10 chomp($line);
11 if ($line=~/.*gene.*;Name=([\w\d\_]+);.*/){
12    print ($fh "$1\n");
13 }
14 }
15 close ($fh);
16 close (IN);
17 exit 0;
```

Listing D.3: *Script* of the Ensembl Solution for Comparison of Genomic Annotation Software

```
1  use warnings;
2  use Carp;
3  use strict;
4
5  use Venn::Chart;
6  use Array::Contains;
7  use Array::Utils qw(:all);
8  use List::Uniq ':all';
9  use Bio::Seq;
10 use Bio::SeqIO;
11
12
13 #!/usr/bin/perl
14 my $gravaDestinoSW1 = $ARGV[1];
15 my $gravaDestinoSW2 = $ARGV[3];
16 my $INTERSEC = "./intersec.txt";
17
18 my $fileonlySW1 = "./onlySW1.txt";
19 my $fileonlySW2 = "./onlySW2.txt";
20
21 my $line;
22 open (IN, $gravaDestinoSW1) or die "cannot open file";
23
24 open(my $fh, '>', $INTERSEC) or die "cannot open file";
```

```
25
26 my @SW1;
27 my @SW2;
28
29 #################
30 #Creating SW1 array
31
32 while ($line = <IN>){
33
34 chomp($line);
35 if ($line=~/(.*)/){
36    print ("$1\n");
37    push(@SW1,$1);
38 }
39
40 }
41
42 close (IN);
43
44 #################
45 #Creating SW2 array
46
47 open (IN2, $gravaDestinoSW2) or die "cannot open file";
48
49 while ($line = <IN2>){
50
51 chomp($line);
52 if ($line=~/(.*)/){
53    print ("$1\n");
54    push(@SW2,$1);
55 }
56
57 }
58
59 close (IN2);
60
61 my @uniqSW1= uniq(@SW1);
62 my @uniqSW2= uniq(@SW2);
63
```

```
64  #################
65  #Only SW1
66
67  my %h;
68
69  @h{@uniqSW2}=undef;
70
71  my @onlySW1 = grep {not exists $h{$_}} @uniqSW1;
72
73  open(my $opp, '>', $fileonlySW1) or die "cannot open file";
74
75  foreach my $op (@onlySW1){
76  print($opp "$op\n");
77  }
78
79  print "(Only ".$ARGV[0].") Genes File created.\n";
80  close ($opp);
81
82  #################
83  #Only SW2
84
85  my %h1;
86
87  @h1{@uniqSW1}=undef;
88
89  my @onlySW2 = grep {not exists $h1{$_}} @uniqSW2;
90
91
92  open(my $omp, '>', $fileonlySW2) or die "cannot open file";
93
94  foreach my $om (@onlySW2){
95  print($omp "$om\n");
96  }
97
98  print "(Only ".$ARGV[2].") Genes File created.\n";
99  close ($omp);
100
101
102 #################
```

```
103 #Creating Intersection set
104
105
106
107 my @INTERSECTION = intersect(@uniqSW1, @uniqSW2);
108
109 foreach my $p (@INTERSECTION){
110 print($fh "$p\n");
111
112 }
113
114
115 print "Intersection File created.\n";
116 close ($fh);
117
118 ################
119 # Create the Venn::Chart constructor
120 my $venn_chart = Venn::Chart->new( 550, 550 ) or die("error : $!");
121
122 # Set a title and a legend for our chart
123 $venn_chart->set_options( -title => 'Venn diagram - Pipelines
       Comparison' );
124 #$venn_chart->set_legends( 'SW1', 'SW2' );
125 $venn_chart->set_legends( $ARGV[0], $ARGV[2] );
126 # 3 lists for the Venn diagram
127
128
129 # Create a diagram with gd object
130 my $gd_venn = $venn_chart->plot( \@uniqSW1, \@uniqSW2 );
131
132 # Create a Venn diagram image in png, gif and jpeg format
133 open my $fh_venn, '>', 'VennChart.png' or die("Unable to create png
       file\n");
134 binmode $fh_venn;
135 print {$fh_venn} $gd_venn->png;
136 close $fh_venn or die('Unable to close file');
137
138 # Create an histogram image of Venn diagram (png, gif and jpeg
       format)
```

```
139 my $gd_histogram = $venn_chart->plot_histogram;
140 open my $fh_histo, '>', 'VennHistogram.png' or die("Unable to
    create png file\n");
141 binmode $fh_histo;
142 print {$fh_histo} $gd_histogram->png;
143 close $fh_histo or die('Unable to close file');
144
145 # Get data list for each intersection or unique region between the
    3 lists
146 my @ref_lists = $venn_chart->get_list_regions();
147 my $list_number = 1;
148 foreach my $ref_region ( @ref_lists ) {
149 # print "List $list_number : @{ $ref_region }\n";
150 $list_number++;
151 }
152 print "Venn diagram created.\n";
153
154 #################
155 #Gererating Reports
156
157
158 my $seqin = Bio::SeqIO->new(-file => $ARGV[4] );
159
160 #$seqout= Bio::SeqIO->new( -format => 'Fasta', -file => '>output.fa
    ');
161
162 my $reportonlySW2 = "./reportonlySW2.txt";
163 my $reportonlySW1 = "./reportonlySW1.txt";
164
165 open(my $rom, '>', $reportonlySW2) or die "cannot open file";
166 open(my $rop, '>', $reportonlySW1) or die "cannot open file";
167
168 my %genomicdata=();
169 my $gene="";
170 my $note="";
171
172 #################
173 #Generates all data
174
```

```
175
176
177 my $seq_object = $seqin->next_seq;
178
179 for my $feat_object ($seq_object->get_SeqFeatures) {
180 if ($feat_object->primary_tag eq "gene") {
181
182     if ($feat_object->has_tag("gene")) {
183         for my $val ($feat_object->get_tag_values("gene")) {
184             #print $val." ";
185         $gene=$val;
186     }
187     }
188     if ($feat_object->has_tag("note")) {
189         for my $val ($feat_object->get_tag_values("note")) {
190             #print $val."\n";
191         $note=$val;
192         }
193     }
194 $genomicdata{$gene}=$note;
195 }
196 }
197
198 ###############
199 #Creating report files
200
201 print ($rom "\t\t\t\t\tExample of Genes found on ".$ARGV[2]."
        pipeline\n\n");
202
203 print ($rop "\t\t\t\t\tExample of Genes found on ".$ARGV[0]."
        pipeline\n\n");
204
205 for (keys %genomicdata){
206 if(contains($_,\@onlySW2)){
207         print ($rom "Gene: ".$_."\t Obs.:\t".$genomicdata{$_}."\n\n
            ");
208 }
209 }
210
```

```
211
212 for (keys %genomicdata){
213 if(contains($_,\@onlySW1)){
214         print ($rop "Gene: ".$_."\t Obs.:\t".$genomicdata{$_}."\n\n
                ");
215 }
216 }
217
218 close $rom;
219
220 close $rop;
221
222 print "Report files created\n";
223
224 exit 0;
```

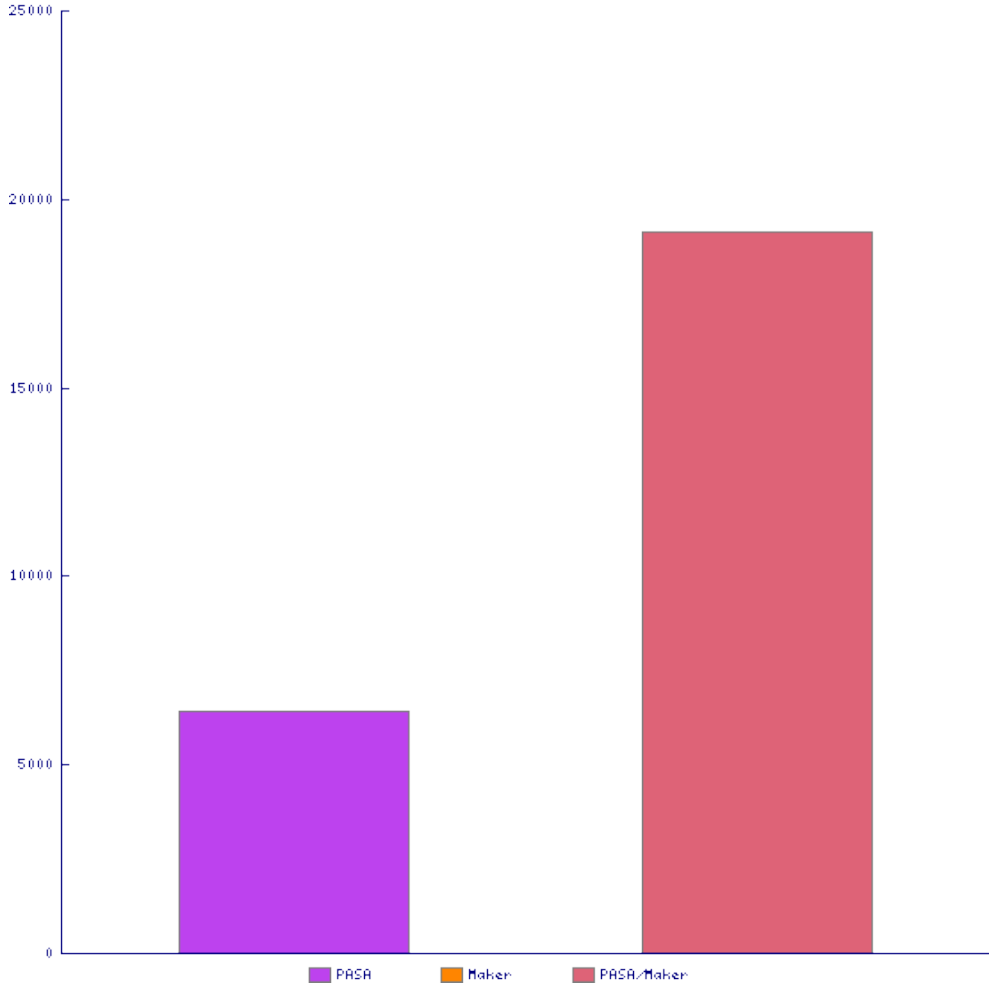

Figura D.1: Histograma de Venn - Número de genes encontrados nas pipelines de *PASA* e *MAKER*

Fonte: Software Ensemble Solution para Comparação de Software de Anotação Genômica

APÊNDICE E – CONFIGURAÇÃO GERAL DE SISTEMA LINUX PARA O SOFTWARE *PASA*

Este HowTo abrange toda a configuração necessária para rodar com sucesso a ferramenta *PASA* num ambiente Linux e foi testado na versão Ubuntu 16.04.

Inicialmente é preciso acessar HAAS (2019a) e obter o software propriamente dito. Além do download via protocolo Http também é possível abrir o prompt de comando, entrar no diretório destinado ao projeto e executar:

Listing E.1: Obtendo o software *PASA*

```
1 git clone https://github.com/PASApipeline/PASApipeline
```

Se ainda não tiver instalado o pacote git, então recomendam-se os seguintes comandos:

Listing E.2: Instalando o Git

```
1 sudo apt-get update
2 sudo apt-get install git
```

Uma vez copiada a estrutura de diretórios, entrar na pasta PASApipeline/Docker com o intuito de executar o *script* **build_Docker.sh** que através da Internet realizará vários downloads e configurações necessárias na máquina.

Contudo para tanto é necessário que uma estrutura Docker esteja instalada. Se ainda não for o caso, serão necessários alguns passos a seguir:

Listing E.3: Instalando o Docker

```
sudo apt-get install docker.io
```

Neste momento ainda não há permissão no sistema para rodar o *script* mencionado. O comando acima criou um grupo de usuários chamado docker então faça:

Listing E.4: Configurando o Docker

```
sudo usermod -aG docker $USER
```

Agora é importante reiniciar o sistema para continuar.

O próximo componente é o blat, cujo binário pode ser encontrado em CRUZ (2019), mais especificamente no link “Blat” e então “utilities directory”.

Após o download do referido arquivo binário, dê permissão de execução para o arquivo:

Listing E.5: Dando Permissão de Acesso ao blat

```
 chmod +x ./blat
```

Para dar visibilidade ao mesmo exporta-se a variável $PATH e adicionando o caminho da pasta onde se encontra o executável do blat. Porém em seguida estes comandos validam as alterações:

Listing E.6: Exportando a variável $PATH

```
export PATH = ''$PATH:/caminhos-diretorios-blat''
source ~/.profile
ou
source ~/.bashrc
```

As seguintes bibliotecas precisam ser instaladas para que o banco de dados funcione corretamente:

Listing E.7: Instalando bibliotecas para banco de dados

```
1 sudo apt-get install libdbi-perl
2 sudo apt-get install libdbd-sqlite3-perl
```

Também deve-se instar o componente *cdbfasta* podendo-se fazer uso da ferramenta de instalação padrão do sistema:

Listing E.8: Instalando componente cdbfasta

```
1 sudo apt-get install cdbfasta
```

Para a geração de gráficos é utilizada a ferramenta R portanto procede-se a instalação desta linguagem da seguinte forma:

Listing E.9: Instalando a linguagem R

```
1 sudo apt-get install r-base r-base-dev
```

Uma opção do software *PASA* é *–TRANSDECODER* e para tanto procede-se a instalação da ferramenta:

Listing E.10: Instalação do Transdecoder

```
1 sudo apt-get install transdecoder
```

Também são importantes os programas *fasta36, fastf36, fasts36, ggsearch36, glsearch36...* entre outros, integrantes do pacote de W. R. Pearson que pode ser encontrado em PEARSON (2019).

Contudo não basta fazer o download para sua utilização pelo programa *PASA*. Para criar os executáveis faça os seguintes passos:

Listing E.11: Instalação dos binários fasta

```
1 cd src
2 make -f ../make/Makefile.linux_sse2 all
```

E então renomeie na subpasta *bin* os arquivos com terminação “36” deixando apenas o nome do arquivo sem o numeral final: por exemplo: fasta36 para apenas fasta.

Por fim adicione esta subpasta *bin* à variável de ambiente $PATH (como indicado no exemplo anterior).

A última instrução se refere à instalação do software GMAP (WU, 2019).

Extraídos os arquivos, execute:

Listing E.12: Instalando o GMAP

```
1 ./configure && make && make install
```

APÊNDICE F – PARÂMETROS DE ENTRADA, UTILIZAÇÃO E RESULTADOS DO SOFTWARE *PASA*

PARÂMETROS DE ENTRADA

Entre os subdiretórios da pasta PASApipeline/ pode-se começar a aprender a utilizá-lo por uma chamada *sample_data*. Nela, o arquivo fasta com os dados genômicos está compactado no formato .gz (genome_sample.fasta.gz) então o primeiro passo é utilizar o comando gunzip para descompactá-lo.

O software pode utilizar-se tanto do banco de dados MySql quanto do SQLite sendo que para o primeiro algumas configurações adicionais devem ser feitas no arquivo PASAPipeline/pasa_conf/conf.txt:

Listing F.1: Configurando uso com MySql

```
1  ######################################
2  ## PASA admin settings ###############
3  ######################################
4
5  #emails sent to admin on job launch, success, and failure
6  PASA_ADMIN_EMAIL=bhaas@tigr.org
7
8  # database to manage pasa jobs; required for daemon-based
       processing.
9  PASA_ADMIN_DB=PASA2_admin_06152006_devel
10 ...
11
12 ######################################
13 ## MySQL settings: ##################
14 ######################################
```

```
15
16 # server actively running MySQL
17 MYSQLSERVER=localhost
18
19 # read-only username and password
20 MYSQL_RO_USER=pasa_access
21 MYSQL_RO_PASSWORD=pasa_access
22
23 # read-write username and password
24 MYSQL_RW_USER=access
25 MYSQL_RW_PASSWORD=access
```

Para criar este arquivo basta copiar e renomear o arquivo já existente sample_test.conf (da mesma pasta).

Na primeira parte constarão o email do adminstrador local e o nome lógico do banco de dados, que inclusive comporá alguns arquivos gerados no processo. Já na segunda tem-se o endereço do servidor e as especificações de dois usuários: MYSQL_RO_USER e MYSQL_RO_PA SSWORD são o login e senha de um usuário com direitos somente-leitura; enquanto que MYSQL_RW_USER e MYSQL_RW_PASSWORD são de um usuário com todos os direitos.

Se a opção for pelo SQLite não é necessário criar o arquivo conf.txt.

Para facilitar existem dois arquivos: runMe.MySQL.sh e runMe.SQLite.sh que iniciam os testes utilizando os respectivos banco de dados. Seu conteúdo é mostrado na Listagem F.2:

Listing F.2: Conteúdo dos arquivos runMe.MySQL.sh

```
1
2 #!/bin/bash
3
4 set -ev
5
6 ./__run_sample_pipeline.pl --align_assembly_config mysql.confs/
    alignAssembly.config --annot_compare_config mysql.confs/
```

```
annotCompare.config
```

Neste caso os parâmetros *–align_assembly_config* e *–annot_compare_config* estão configurados para a pasta do mysql.confs/ onde a variável DATABASE dos arquivos alignAssembly.config e annotCompare.config são iguais ao nome do banco de dados que deve estar no arquivo conf.txt.

Se optar pelo SQLite basta rodar o *script* runMe.SQLite.sh lembrando-se de configurar a variável DATABASE dos arquivos sqlite.confs/alignAssembly.config e sqlite.confs/ annotCompare.config para o nome do banco de dados.

Os dados transcritos se encontram nos arquivos *all_transcripts.fasta, all_transcripts.fasta. clean e all_transcripts.fasta.cln* mas o arquivo original é o fasta e este é submetido a um processo de limpeza através do software *seqclean* e então os demais arquivos são gerados.

Uma vez obtido o arquivo de transcriptoma, deve-se renomeá-lo para *all_transcripts.fasta* e submetê-lo ao processo de limpeza:

Listing F.3: Iniciando processo de limpeza dos transcritos

```
1 ../bin/seqclean all_transcritps.fasta -cX
```

Onde X é o número de CPUs determinadas a serem utilizadas no processo. Os demais arquivos são aqui gerados.

UTILIZAÇÃO

Então tudo está pronto para a execução do *script* que inicia a pipeline, será utilizado *./runMe.SQLite.sh* (semelhante à Figura F.2). E o primeiro comando internamente executado é:

Listing F.4: Comando inicial da pipeline do *PASA*

```
1 ../Launch_PASA_pipeline.pl -c $align_assembly_config_file -C -r -R
    -g genome_sample.fasta -t all_transcripts.fasta.clean -T -u
    all_transcripts.fasta -f FL_accs.txt --ALIGNERS $ALIGNERS --CPU
    $CPU -N $num_top_hits --TDN tdn.accs  --
```

```
   IMPORT_CUSTOM_ALIGNMENTS_GFF3 custom_alignments.gff3
```

O parâmetro $align_assembly_config_file remete ao subdiretório correspondente ao banco de dados e ao arquivo alignAssembly.config escolhidos quando da invocação do script. $ALIGNERS está configurado por default para "gmap,blat" e $CPU também por default é 2 e representa o número de CPUs que serão empregadas no processamento.

A opção -f FL_accs.txt não é obrigatória, podendo ser substituída por -f NULL, assim como podem ser completamente omitidas –TDN tdn.accs e –IMPORT_CUSTOM_ALIGNMENT S_GFF3 custom_alignments.gff3.

As opções *-C -r -R* definem que se o banco de dados já existir, ele será recriado e naturalmente populado nesta nova execução.

Este processo demanda algum tempo e em breve serão discutidos os resultados. O próximo comando executado será:

Listing F.5: Building comprehensive transcriptome

```
1 ../scripts/build_comprehensive_transcriptome.dbi -c $
     align_assembly_config_file -t all_transcripts.fasta.clean
```

Logo após segue a primeira rodada de comparações, usando anotações de estruturas gênicas pré-existentes:

Listing F.6: Primeira rodada de comparações

```
1 ../Launch_PASA_pipeline.pl -c $annot_compare_config_file -g
     genome_sample.fasta -t all_transcripts.fasta.clean -A -L --
     annots orig_annotations_sample.gff3 --CPU $CPU
```

Para obter o arquivo *orig_annotations_sample.gff3* basta copiar o arquivo em formato GFF3 por exemplo retirado da plataforma NCBI e trocar seu nome.

Depois viria uma segunda rodada de comparações já com as anotações criadas pelo próprio *PASA* em formato GFF3 num arquivo chamado *$PASA_dbname.gene_st*

PASA_updates.XXXX.gff3.

Listing F.7: Segunda rodada de comparações

```
1 ../Launch_PASA_pipeline.pl -c $annot_compare_config_file -g
     genome_sample.fasta -t all_transcripts.fasta.clean -A -L --
     annots $recent_update_file --CPU $CPU
```

Nas experiências realizadas, o arquivo citado no parágrafo anterior resultou vazio e esta parte do processo quebrou a execução da pipeline com a mensagem de erro que este arquivo não era um arquivo de formato válido mas na realidade estava vazio pois provavelmente nenhuma alteração foi percebida pelo *PASA*.

Para dar continuidade no processo, buscou-se no código-fonte do arquivo *__run_sample_ pipeline.pl* o próximo comando que se refere ao registro de *SPLICING* alternativo:

Listing F.8: Registrando Splicing alternativo

```
1 ../Launch_PASA_pipeline.pl -c $annot_compare_config_file -g
     genome_sample.fasta -t all_transcripts.fasta.clean --CPU $CPU --
     ALT_SPLICE
```

Por fim há um procedimento para encontrar Open Reading Frames (ORFs) cujo início se dá pelo comando:

Listing F.9: Procurando ORFs

```
1 ../scripts/pasa_asmbls_to_training_set.dbi --pasa_transcripts_fasta
      $DBname.assemblies.fasta --pasa_transcripts_gff3 $DBname.
     pasa_assemblies.gff3
```

Aqui termina o *script* de exemplo que vem com o software *PASA*, contudo é a outro comando semelhante a este que se devem os relatórios gráficos em formato PDF interessantes como por exemplo as curvas ROC:

Listing F.10: Extraindo informações na forma de gráficos

```
../scripts/pasa_asmbls_to_training_set.extract_reference_orfs.pl
   best_candidates.gff3 [minProtLength=300]
```

Finalizado o processamento, o conjunto de arquivos gerados (exceto obviamente os iniciais) é o seguinte:

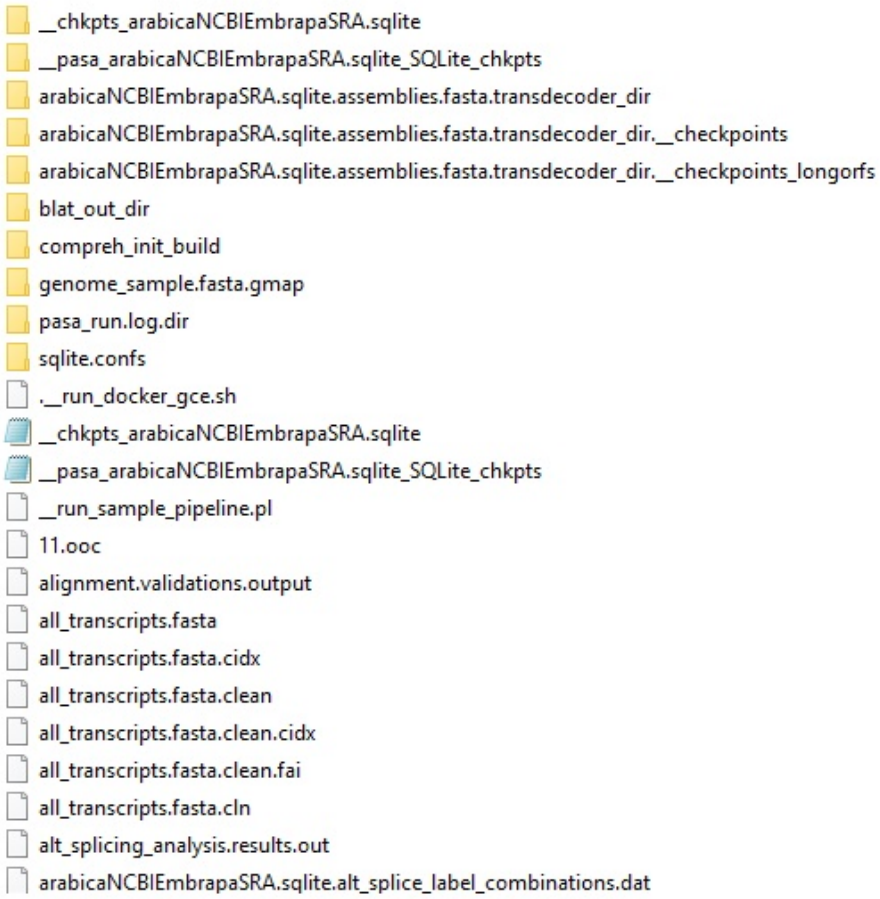

Figura F.1: Listagem de arquivos gerados pelo *PASA* - A

Fonte: O Autor

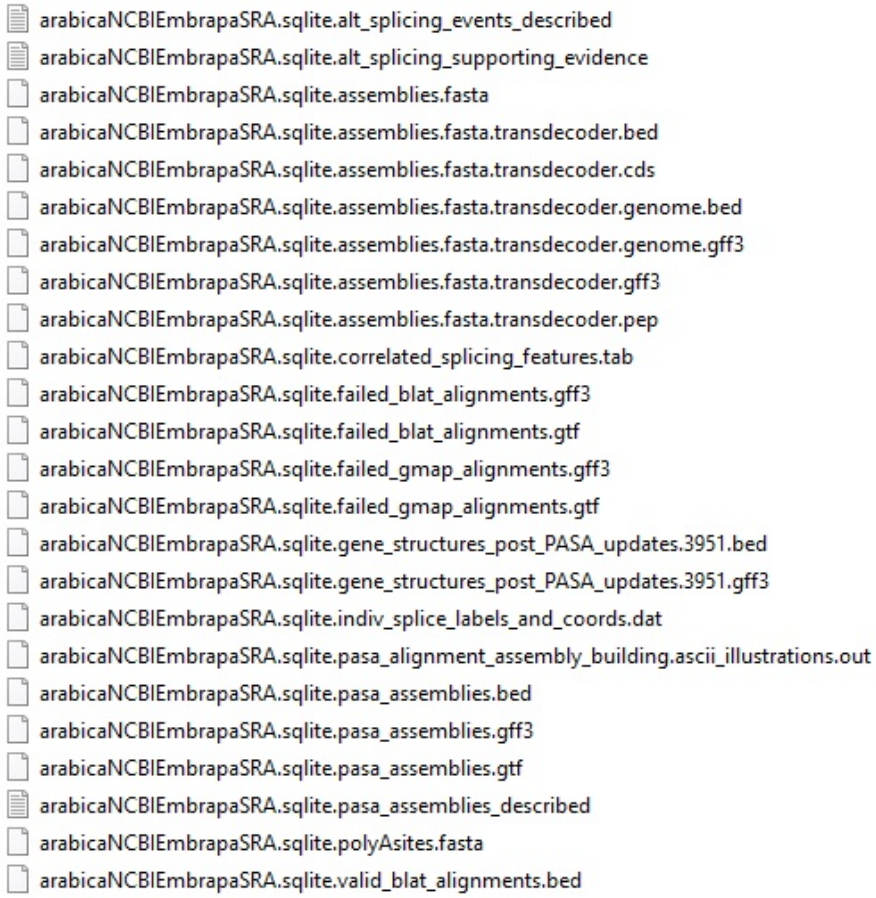

Figura F.2: Listagem de arquivos gerados pelo *PASA* - B

Fonte: O Autor

Figura F.3: Listagem de arquivos gerados pelo *PASA* - C

Fonte: O Autor

Já dentro da pasta *compreeh_init_build* tem-se:

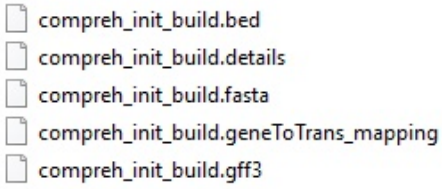

Figura F.4: Conteúdo de compreeh_init_build

Fonte: O Autor

O subdiretório que produz gráficos em formato PDF utilizando recursos da ferramenta R é o seguinte:

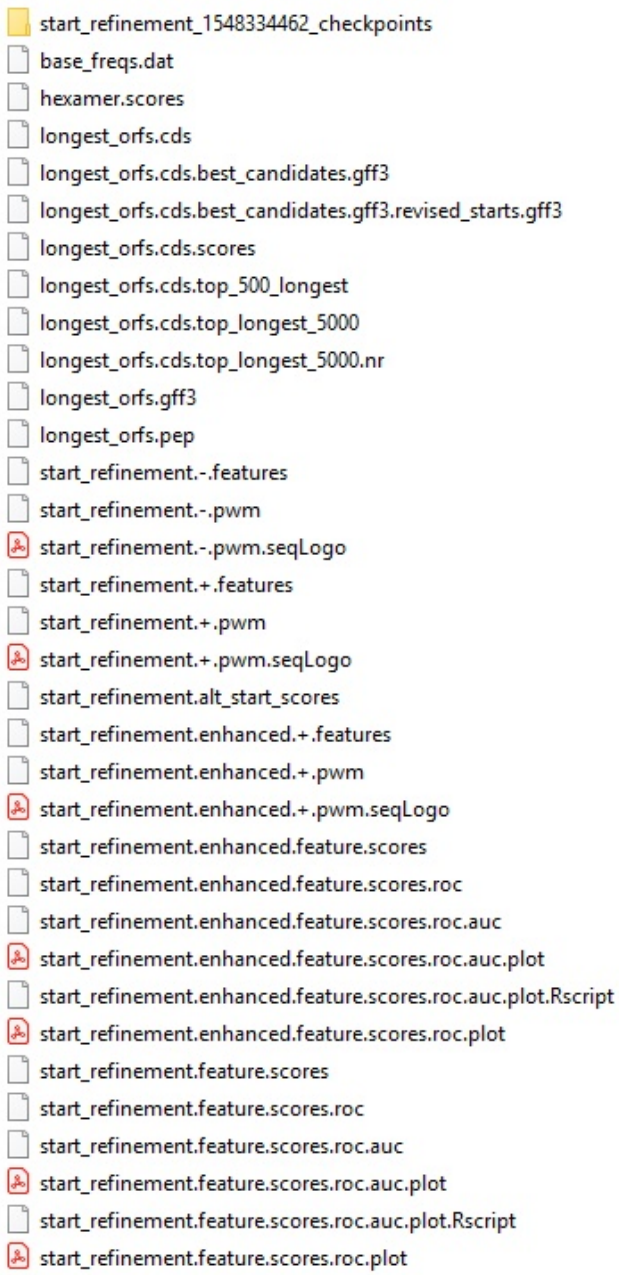

Figura F.5: Conteúdo de $DBname.assemblies.fasta.transdecoder_dir

Fonte: O Autor

Os subdiretórios cujos nomes possuem *chkpts* ou mesmo *checkpoints* não são aqui listados pois tem apenas papel de checkagem para o software. Seguem aqueles de maior relevância:

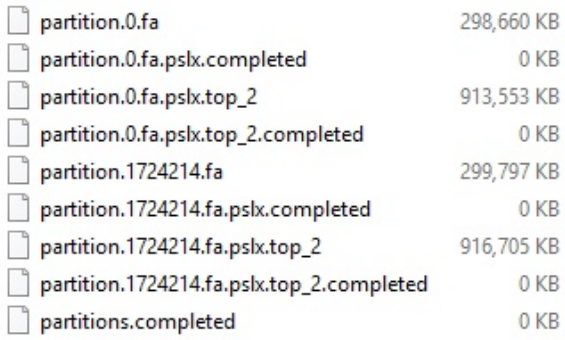

Figura F.6: Conteúdo de blat_out_dir

Fonte: O Autor

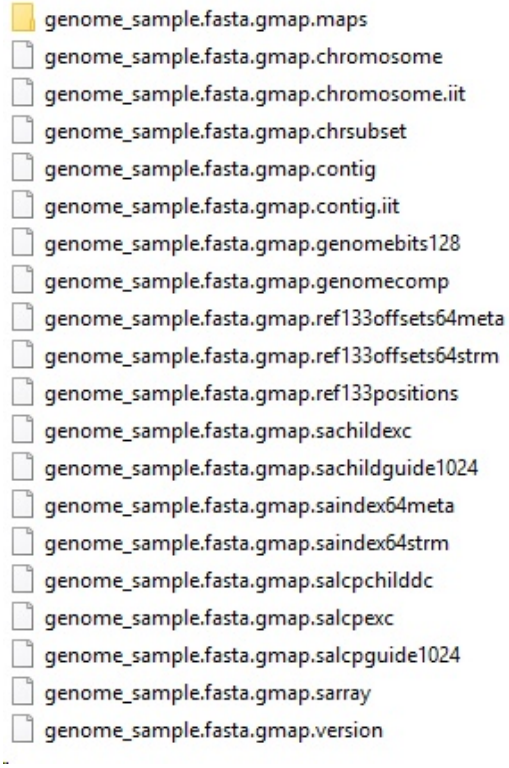

Figura F.7: Conteúdo de genome_sample.fasta.gmap

Fonte: O Autor

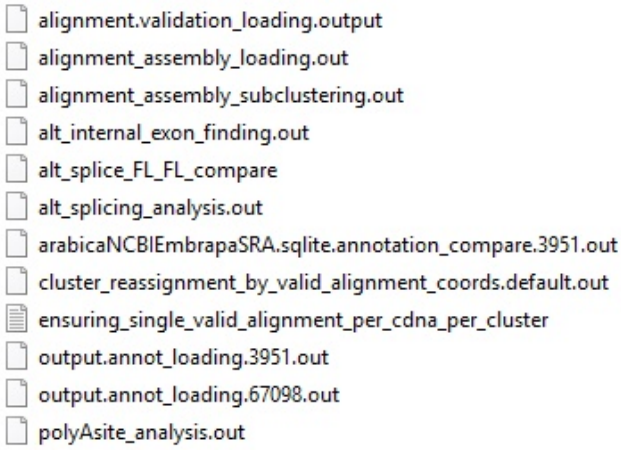

Figura F.8: Conteúdo de pasa_run.log.dir

Fonte: O Autor

Seguem-se os gráficos resultantes do processo, aqueles que representam maior relevância para este trabalho se encontram comentados no corpo principal da dissertação:

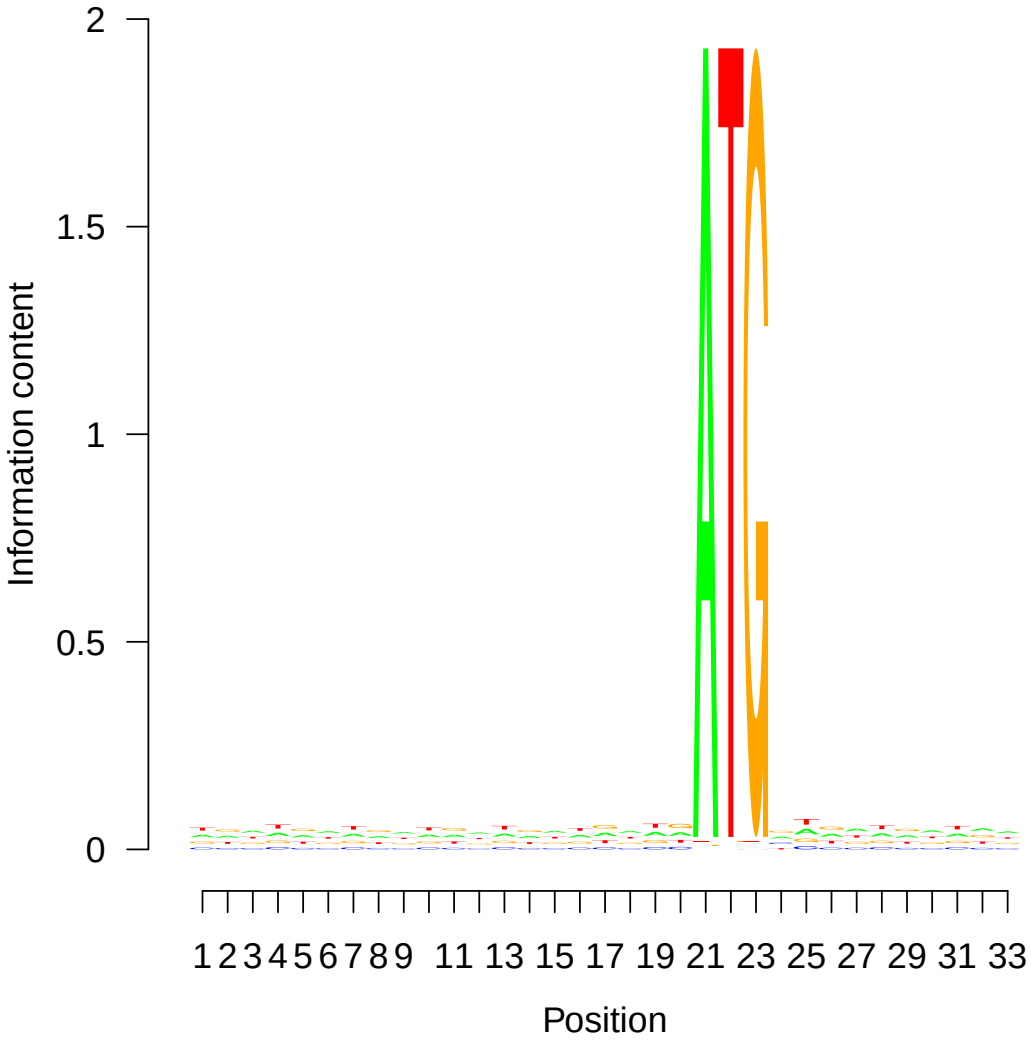

Figura F.9: Gráfico gerado pelo comando start_refinement.-.pwm.seq- Logo

Fonte: Pipeline do *PASA*

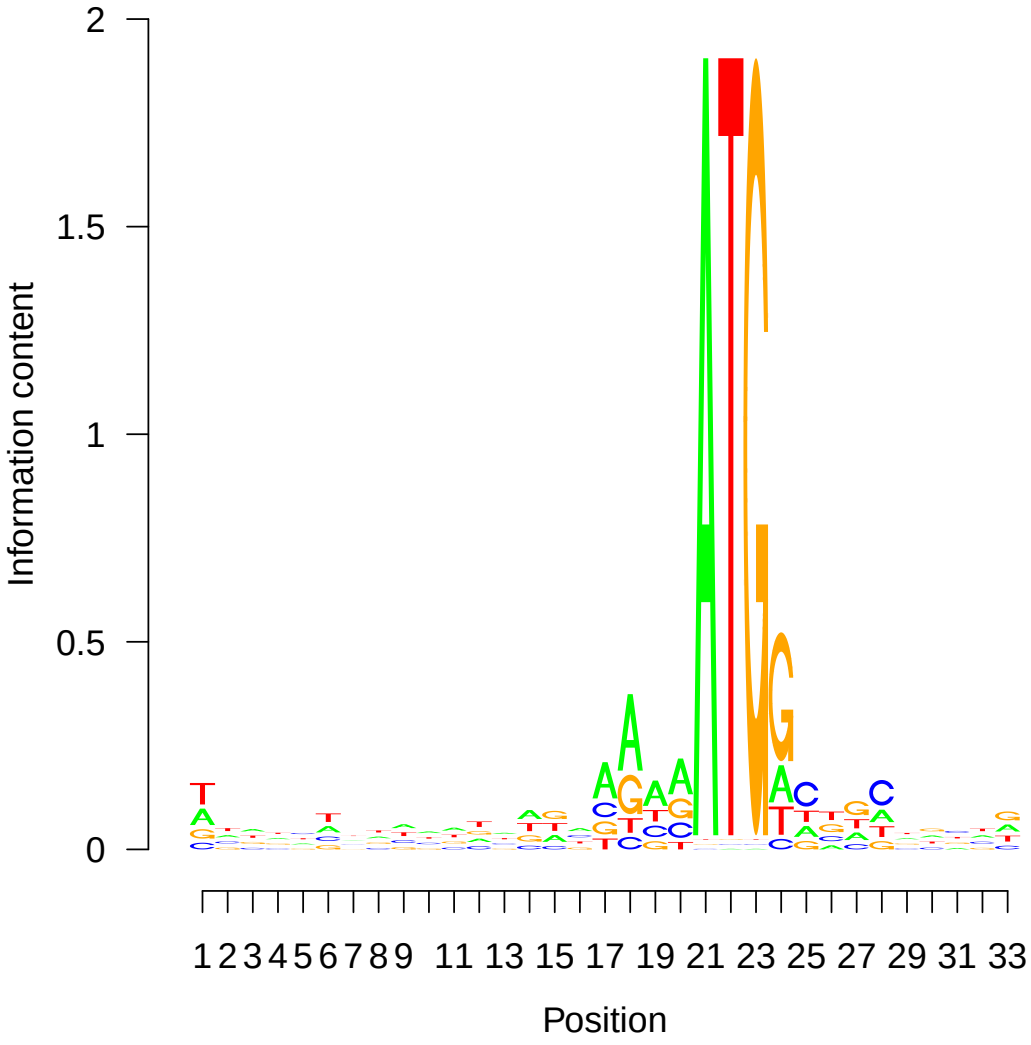

Figura F.10: Gráfico gerado pelo comando start_refinement.+.pwm.seq- Logo

Fonte: Pipeline do *PASA*

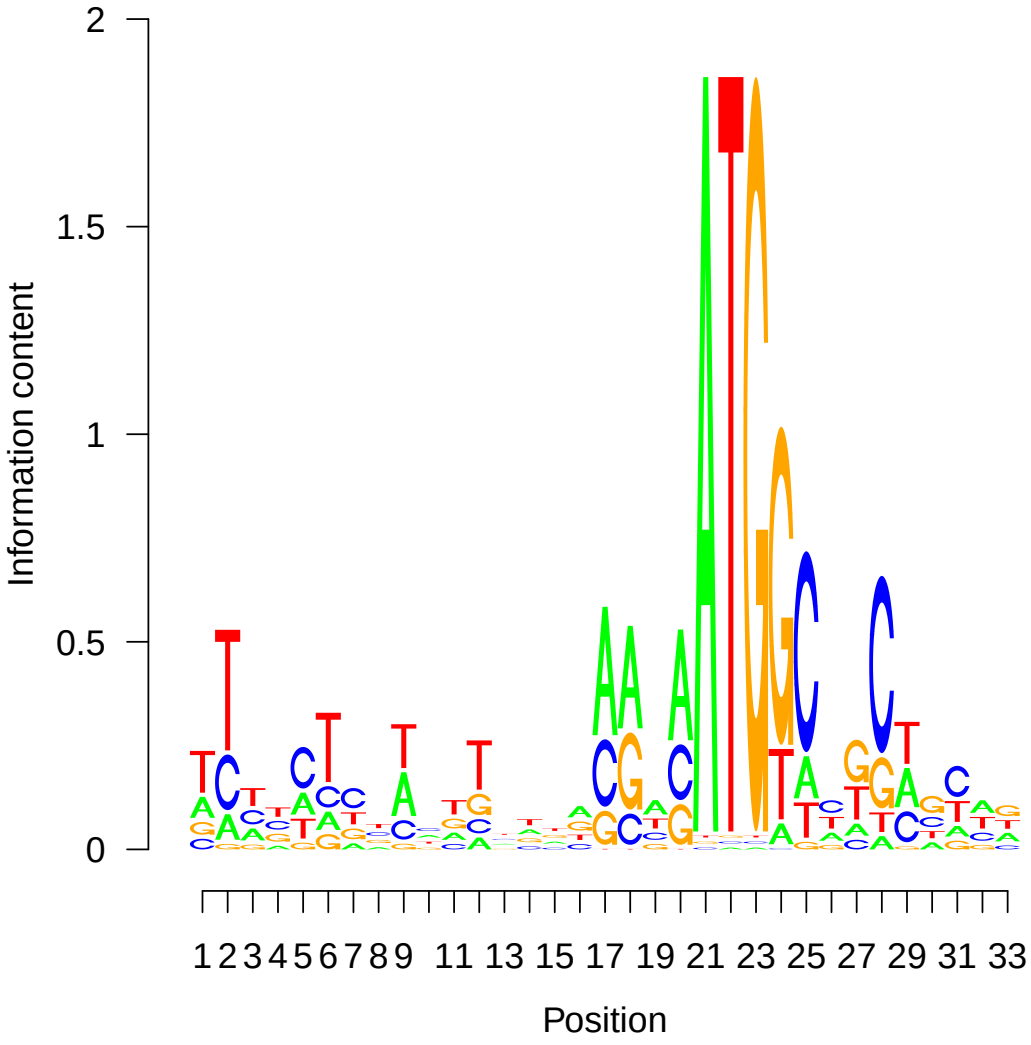

Figura F.11: Gráfico gerado pelo comando start_refinement.enhanced.+.pwm.seqLogo

Fonte: Pipeline do *PASA*

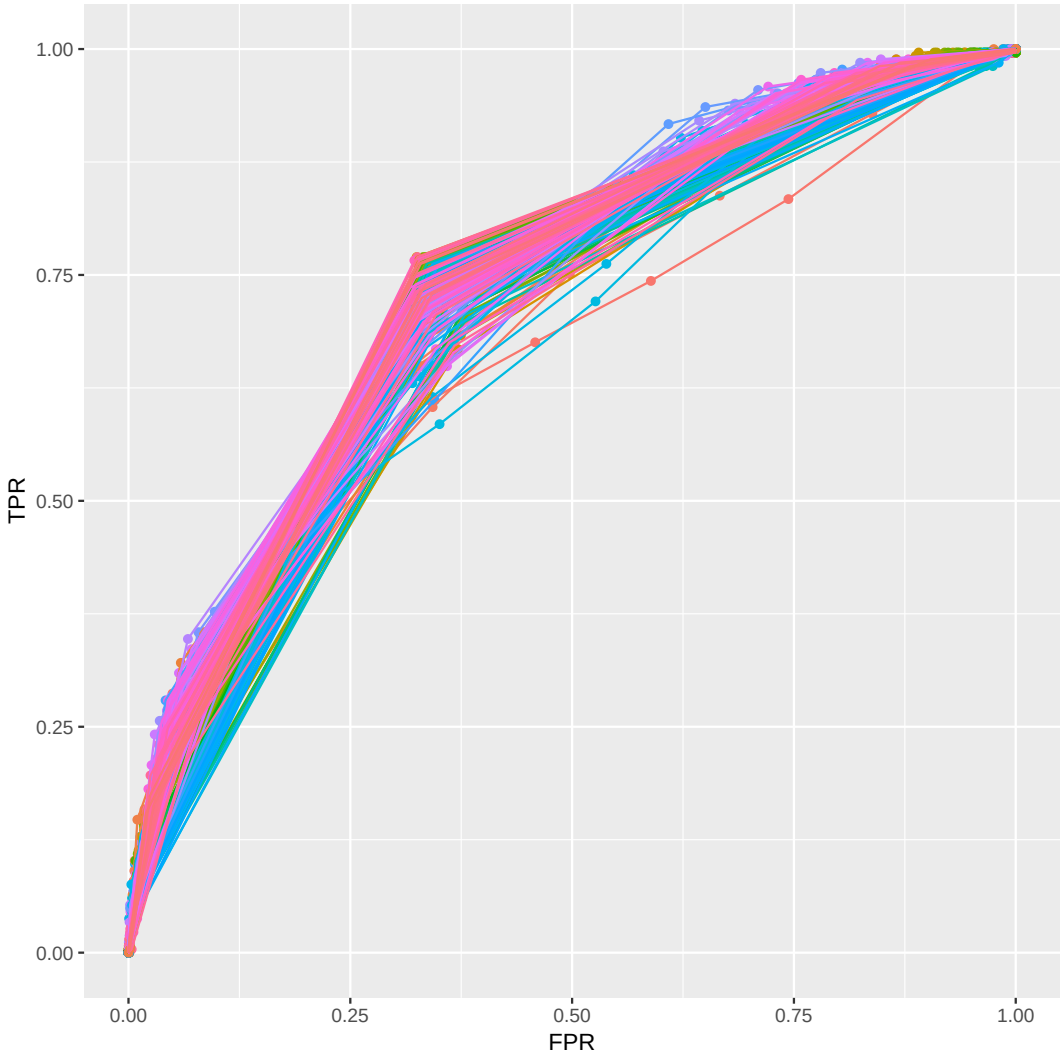

Figura F.12: Gráfico gerado pelo comando start_refinement.feature.scores.roc.plot

Fonte: Pipeline do *PASA*

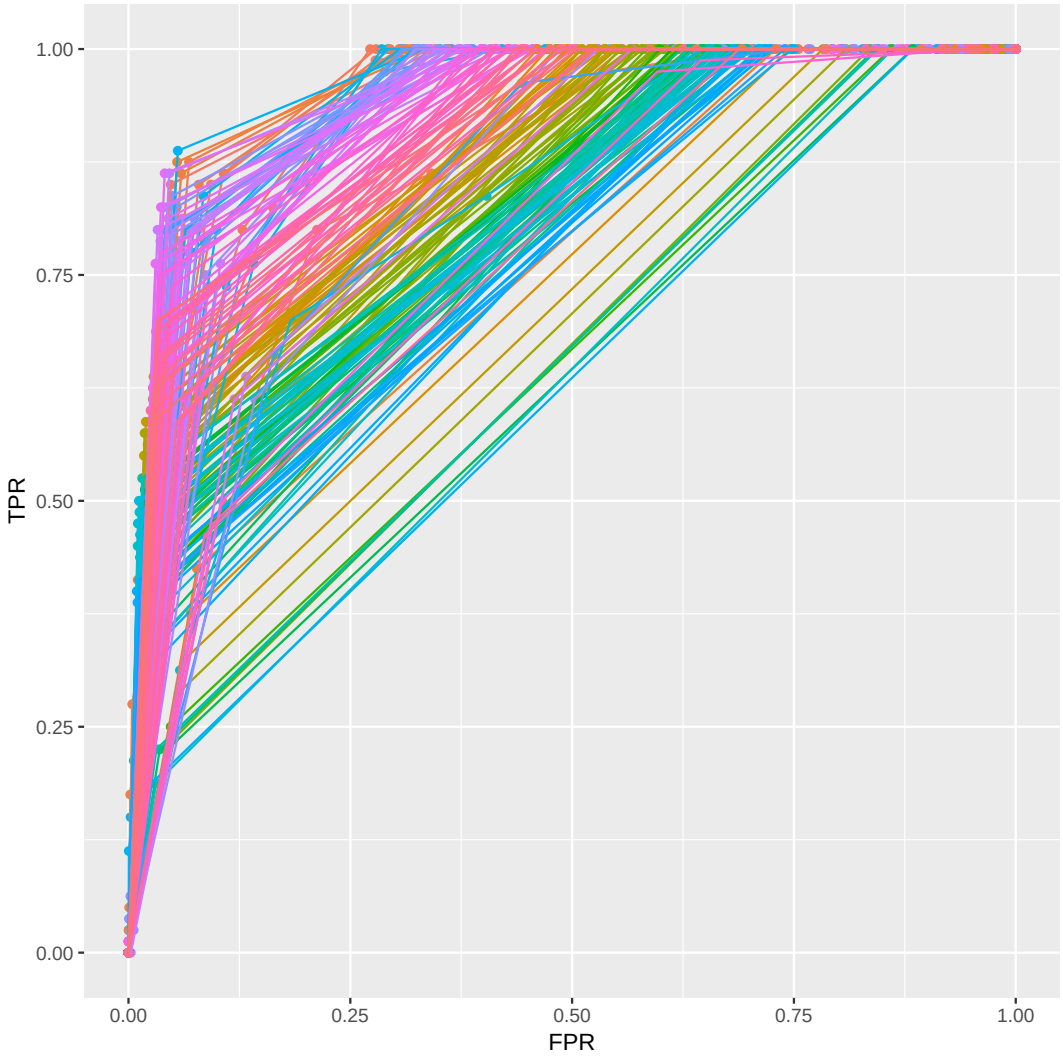

Figura F.13: Gráfico gerado pelo comando start_refinement.enhanced. feature.scores.roc.plot

Fonte: Pipeline do *PASA*