Erinaldo Sanches Nascimento

Ferramenta Web para Descoberta e Categorização de Genes *cry*

Cornélio Procópio

2017

Erinaldo Sanches Nascimento

Ferramenta Web para Descoberta e Categorização de Genes *cry*

Dissertação apresentada ao Programa de Pós-Graduação em Bioinformática como requisito parcial para a obtenção do título de Mestre em Bioinformática pela Universidade Tecnológica Federal do Paraná - Campus Cornélio Procópio.

Universidade Tecnológica Federal do Paraná

Mestrado Acadêmico em Bioinformática

PPGBIOINFO - Programa de Pós Graduação em Bioinformática

Orientador: Prof. Dr. Alessandro Botelho Bovo

Coorientador: Prof. Dr. Laurival Antônio Vilas Boas

Cornélio Procópio

2017

Dados Internacionais de Catalogação na Publicação

N244 Nascimento, Erinaldo Sanches

Ferrramenta web para descoberta e categorização de genes cry / Erinaldo Sanches Nascimento. – 2018.
107 f. : il. color. ; 31 cm.

Orientador: Alessandro Botelho Bovo.
Coorientador: Laurival Antonio Vilas-Boas.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós – Graduação em Bioinformática. Cornélio Procópio, 2018.
Bibliografia: p. 95-99.

1. Análise cladística. 2. Bactérias - classificação. 3. Pragas agrícolas - controle. 4. Bioinformática – Dissertações. I. Bovo, Alessando Botelho, orient. II. Vilas Boas, Laurival Antônio, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Mestrado em Bioinformática. IV. Título.

CDD (22. ed.)572.80285

Biblioteca da UTFPR - Câmpus Cornélio Procópio
Bibliotecários/Documentalistas responsáveis:
Simone Fidêncio de Oliveira Guerra – CRB-9/1276
Romeu Righetti de Araujo – CRB-9/1676

Título da Dissertação Nº 03:

"DESENVOLVIMENTO DE UMA FERRAMENTA WEB PARA DESCOBERTA E CATEGORIZAÇÃO DE GENES *CRY*".

por

Erinaldo Sanches Nascimento

Orientador: Prof. Dr. Alessandro Botelho Bovo

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM BIOINFORMÁTICA – Linha de Pesquisa: Biologia Computacional e Sistêmica, pelo Programa de Pós-Graduação em Bioinformática – PPGBIOINFO – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 09h 00min do dia 27 de setembro de 2017. O trabalho foi _____ pela Banca Examinadora, composta pelos professores:

_____
Prof. Dr. Alessandro Botelho Bovo
(Presidente)

_____
Prof. Dr. Alexandre Rossi Paschoal
(UTFPR-CP)

_____
Profa. Dra. Gislayne Trindade Vilas-Boas
(UEL-LD)

Visto da coordenação:

_____
André Yoshiaki Kashiwabara
Coordenador do Programa de Pós-Graduação em Bioinformática
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

Dedido este trabalho à minha esposa Dálkia Andréia Vilas Boas, que soube compreender minha ausência na confecção do mesmo, e aos meus pais Erinaldo Barboza do Nascimento (in memorian) e Nilza Nogueira Sanches Nascimento, por tudo que me proporcionaram.

AGRADECIMENTOS

Primeiramente agradeço a Deus por me conceder o dom da vida, a saúde, a família que me tem dado e pela força e dedicação para escrever esse trabalho. Ao meus orientadores Prof. Dr. Alessandro Botelho Bovo e Prof. Dr. Laurival Antonio Vilas-Boas que acreditaram em minha capacidade de realização, pelo apoio, incentivo e direcionamentos no âmbito da pesquisa. Agradeço aos amigos, sempre presentes, Cynara, Fábio, Jader, Juliana e Tatiane, que colaboraram unânimes nas disciplinas e eventos que participamos. Um agradecimento especial a todos os professores que tornaram possível, através de suas aulas, a aquisição do conhecimento necessário, tanto na parte biológica quanto computacional, para a execução desse trabalho - Prof. Dr. Fabrício Martins Lopes, Prof. Dr. Alexandre Rossi Paschoal, Profa Dra. Francismar Corrêa Marcelino-Guimarães, Prof. Dr. André Yoshiaki Kashiwabara, Prof. Dr. Laurival Antonio Vilas-Boas, Prof. Dr. Alessandro Botelho Bovo e Prof. Dr. Carlos Nascimento Silla Junior. A Msc. Katia Brumatti Gonçalves da UEL pela ajuda na compreensão e aquisição da base de dados que ela desenvolveu. Ao Msc. Ivan Rodrigo Wolf, por compartilhar o *script* que criou. A Dra. Ana Paula Scaramal Ricietto, que forneceu os resultados de sua pesquisa, com os quais foi possível atestar a eficiência do trabalho proposto. A Profª. Dra. Gislayne Trindade Vilas-Boas presente na banca examinadora, e pelos conselhos fundamentais após a qualificação. Agradeço também a toda equipe de profissionais da UTFPR do campus de Cornélio Procópio pelo incentivo a ciência e tecnologia. Impossível não lembrar dos momentos que desfrutei da boa companhia, das conversas e a hospitalidade na casa do tio Mário e da tia Dirce durante o percurso desse trabalho.

"Tudo quanto vive, vive porque muda; muda porque passa; e, porque passa, morre. Tudo quanto vive perpetuamente se torna outra coisa, constantemente se nega, se furta à vida."

(Fernando Pessoa)

RESUMO

O *Bacillus thuringiensis* (Bt) é uma bactéria formadora de esporos que produz as toxinas Cry, Cyt e Vip, como cristais parasporais, que têm demonstrado serem eficazes no controle de pragas agrícolas e mosquitos vetores de doenças. Os genes codificantes dessas toxinas têm sido usados no desenvolvimento de plantas transgênicas resistentes a insetos. A adoção do biopesticida Bt permite a redução no uso de inseticidas sintéticos, e não oferece riscos ou danos à saúde humana. No entanto, muitas pragas de insetos não são suscetíveis a tais toxinas já identificadas. Até agora, foram identificados mais de 700 genes de Bt relacionados à toxina Cry, alvo nesse trabalho, classificados em mais de 70 grupos. Uma alternativa para as pragas que não são suscetíveis às toxinas Cry parentais é o isolamento de outras cepas de Bt com novos genes *cry* com maior toxicidade, bem como a identificação das moléculas receptoras e epitopos de ligação e a triagem de novas proteínas Cry com toxicidade para novos insetos. Outra alternativa é a evolução genética *in vitro* de tais toxinas. Para auxiliar no processo de encontrar novos genes *cry* foi desenvolvida uma base de dados curada de genes *cry* e implementada uma ferramenta *web* para a identificação e a categorização de uma determinada sequência alvo como pertencente a uma família de gene *cry*. Todo o processo de classificação e categorização combina programas de bioinformática como HMMER, BEDTools, MUSCLE e BLAST. A ferramenta apresenta ao usuário uma lista das sequências alvo com maior identidade para ser um gene *cry*. O usuário seleciona uma dessas sequências para analisar o alinhamento com sequências públicas de genes de Bt. O mesmo procedimento permite reconstruir a árvore filogenética por intermédio de uma matriz de similaridade para identificar os parentes mais próximos. Esse processo pode ser repetido para todas as sequência disponíveis. Os resultados apontaram que a ferramenta tem a capacidade de apoiar o usuário na tarefa de identificar proteínas Cry, com base na sequência de DNA, visando descrever uma proteína já existente ou uma nova proteína Cry, que possa apresentar maior toxicidade e ser empregada para atuar no controle de pragas e vetores de doenças, em um amplo espectro de ação, atingindo as que atualmente não são sensíveis às toxinas Cry ou que são controladas de forma ineficiente.

Palavras-chave: *Bacillus thuringiensis*, gene *cry*, classificação, análise filogenética.

ABSTRACT

Bacillus thuringiensis (Bt) is a spore-forming bacterium that produces Cry, Cyt and Vip toxins as parasporal crystals, which have demonstrated to be effective in controlling agricultural pests and mosquito vectors of diseases. The genes encoding these toxins have been used in the development of insect-resistant transgenic plants. The adoption of Bt biopesticide allows the reduction in the use of synthetic insecticides, and does not offer risks or damages to human health. However, many insect pests are not susceptible to such already identified toxins. So far, more than 700 Cry toxin-related Bt genes have been identified, targeted in this work, classified into more than 70 groups. An alternative for pests that are not susceptible to parental Cry toxins is the isolation of other Bt strains with new *cry* genes with higher toxicity, as well as the identification of receptor molecules and binding epitopes and the screening of novel Cry proteins with toxicity to new insects. Another alternative is the *in vitro* genetic evolution of such toxins. To aid in the process of finding new *cry* genes, a cured *cry* gene database has been developed and a web tool has been implemented for the identification and categorization of a particular target sequence as belonging to a cry gene family. The entire classification and categorization process combines bioinformatics programs such as HMMER, BEDTools, MUSCLE and BLAST. The tool presents the user with a list of the target sequences with the highest identity to be a *cry* gene. The user selects one of these sequences to analyze the alignment with public sequences of Bt genes. The same procedure allows to reconstruct the phylogenetic tree through a matrix of similarity to identify the closest relatives. This process can be repeated for all available sequences. The results pointed out that the tool has the ability to support the user in the task of identifying Cry proteins, based on the DNA sequence, in order to describe an existing protein or a new Cry protein, which may present higher toxicity and be employed to act in the pest control and disease vectors in a broad spectrum of action, reaching those currently not sensitive to Cry toxins or that are inefficiently controlled.

Keywords: *Bacillus thuringiensis*, *cry* gene, classification, phylogenetic analysis.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ALP	*Fosfatase alcalina.*
APN	*Aminopeptidase N.*
ASCII	*American Standard Code for Information Interchange*
BLAST	*Basic Local Alignment Search Tool.*
Bt	*Bacillus thuringiensis.*
Btg	*Bacillus thuringiensis* var. *galleriae.*
Bti	*Bacillus thuringiensis* var. *israelenses*
C-terminal	*Região carboxilo terminal de uma proteína.*
CAD	*Caderina.*
cry	*Corresponde aos genes Bt.*
Cry	*Corresponde aos genes Bt.*
Cyt	*Corresponde aos genes da família citolítica.*
CSV	*Comma-Separated Values.*
DNA	*Ácido desoxirribonucléico.*
EBI	*European Bioinformatics Institute.*
EMBL	*European Molecular Biology Laboratory.*
EPA	*Environment Protection Agency.*
EUA	*Estados Unidos da América.*
GEP	*Gap Extension Penalty.*
GOP	*Gap Opening Penalty.*
GCG	*Genetics Computer Group.*
GPL	*General Public License.*
HMM	*Hidden Markov Models.*

HTTP	*Hyper Text Transfer Protocol.*
HPC	*High Performance Computing.*
IS	*Sequências de inserção.*
JSON	*JavaScript Object Notation.*
kb	*Quilobase* - 10^3 *pares de base.*
kDa	*Quilo daltons (1.000 Daltons).*
MSV	*Multiple Segment Viterbi.*
MUSCLE	*Multiple Sequence Comparison by Log-Expectation.*
N-terminal	*Região amino-terminal de uma proteína.*
NBRF	*National Biomedical Research Foundation.*
NCBI	*National Center for Biotechnology Information.*
NGS	*Next-generation sequencing.*
OECD	*Organization for Economic Cooperation and Development.*
OMS	*Organização Mundial da Saúde.*
ORF	*Open reading frame.*
OTU	*Operational Taxonomic Unit.*
pb	*Pares de base.*
PCR	*Reação em cadeia da DNA Polimerase.*
PFT	*Pore-forming toxins.*
pH	*Potencial hidrogeniônico.*
PIR	*Protein Information Resource.*
REST	*Representational State Transfer.*
RBF	*Radial basis function.*
RNA	*Molécula de ácido nucléico.*
RLO	*Radicais livres de oxigênio.*
SOAP	*Simple Object Access Protocol.*

SSV	*Single ungapped Segment Viterbi.*
SVM	*Support Vector Machine.*
Tn	*Transposons.*
TSV	*Tab-Separated Values.*
UPGMA	*Unweighted Pair Group Method with Arithmetic.*
UV	*Raios ultravioleta.*
Vip	*Proteína inseticida vegetativa.*
XML	*eXtensible Markup Language.*

SUMÁRIO

1 INTRODUÇÃO

Os danos à colheita devido a insetos, fungos, bactérias e vírus podem ser responsáveis por 35% das perdas da produção agrícola mundial. O método comum de controle depende do uso de inseticidas sintéticos. Porém, o uso contínuo dos mesmos pode levar ao desenvolvimento de resistência entre as populações de insetos, tornando esses produtos ineficientes no controle de pragas. Adicionalmente deve-se considerar que o controle de pragas na agricultura e de insetos vetores de doenças usando inseticidas sintéticos, além de poluirem o meio ambiente, aumentam os riscos à saúde humana podendo levar ao desenvolvimento de câncer e vários distúrbios do sistema imunológico (PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; MONNERAT et al., 2006; BRAVO et al., 2011).

Bacillus thuringiensis (Bt) é uma bactéria de grande interesse agronômico, científico e importante na saúde pública, capaz de fazer o controle microbiano e é utilizada como substitutivo para os produtos sintéticos. Bt é uma bactéria aeróbica, ubíqua, Gram-positiva e formadora de esporos. As células vegetativas possuem forma de bastonete e sintetizam inclusões cristalinas parasporais compostas por δ-endotoxinas, ou toxinas Cry, as quais são ativas contra larvas de uma variedade de espécies de insetos. Sendo, portanto, usadas no controle biológico de insetos. A bioatividade específica de Bt é causada pela produção desses cristais predominantemente compostos por uma ou mais proteínas inseticidas conhecidas como toxinas Cry (SCHNEPF et al., 1998; MAAGD; BRAVO; CRICKMORE, 2001; ROH et al., 2007; NOGUERA; IBARRA, 2010; BRAVO et al., 2011; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; PALMA et al., 2014; ADANG; CRICKMORE, 2014).

O uso de Bt no controle de pragas de insetos agrícolas, pragas desfolhadoras e vetores de doenças tem resultado em uma alternativa suplementar significativa à aplicação de pesticidas sintéticos na agricultura comercial e no controle de mosquitos. Outro avanço importante na redução de inseticidas sintéticos na agricultura é o desenvolvimento de culturas transgênicas capazes de expressar toxinas Cry (MAAGD; BRAVO; CRICKMORE, 2001; MONNERAT et al., 2006; ROH et al., 2007; BRAVO et al., 2011; YE et al., 2012; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; PALMA et al., 2014).

Historicamente, o Bt foi isolado pela primeira vez no Japão em 1901 por Shigetane Ishiwatari, em larvas do bicho da seda doentes. A atividade inseticida da proteína Cry levou ao desenvolvimento global de bioinseticidas baseados em Bt para o controle de pragas. O uso desses bioinseticidas na agricultura remonta a aproximadamente 70 anos, quando se tornou disponível na França (ROH et al., 2007; VILAS-BOAS; PERUCA; ARANTES, 2007; BRAVO et al., 2013).

Em 1977 foi descoberto uma subespécie de Bt, a var. *israelenses* (Bti), que é

conhecida por apresentar alta toxicidade ao *Aedes aegypti*, vetor da dengue, da febre amarela, do vírus Zika e da febre chikungunya. Os produtos à base de Bti foram utilizados na Alemanha, na China e na África Ocidental para controle de mosquitos, vetores de malária e oncocercose, respectivamente. De 1982 a 1997 foram usados mais de 5 milhões de litros de Bti no controle dessas pragas. Em 1995 a venda de produtos em escala mundial foi projetada em torno de 90 milhões de dólares, representando um total de 2% do mercado global de inseticidas. Em 1996, através da biotecnologia agrícola, variedades de batata, algodão e milho contendo genes *cry* foram vendidas aos produtores. Em 1998 foram registrados 200 produtos de Bt nos EUA. Em 2009 mais de 40 milhões de hectares foram cultivados com culturas Bt em escala mundial. Já em 2010 foram mais de 58 milhões de hectares cultivados, principalmente com culturas transgênicas de milho-Bt ou de algodão-Bt (SCHNEPF et al., 1998; MAAGD; BRAVO; CRICKMORE, 2001; VILAS-BOAS; PERUCA; ARANTES, 2007; ROH et al., 2007; BRAVO et al., 2013; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; ADANG; CRICKMORE, 2014).

Linhagens de Bt produzem também outras toxinas, sendo que até hoje já foram isoladas e classificadas mais de 700 sequências de genes de Bt pertencentes à família de genes *cry* e outras duas famílias de proteínas com atividade entomopatogênica não filogeneticamente relacionadas (MAAGD; BRAVO; CRICKMORE, 2001; VILAS-BOAS; PERUCA; ARANTES, 2007; BRAVO et al., 2011; YE et al., 2012; BRAVO et al., 2013; PALMA et al., 2014).

As estirpes de Bt podem ser encontradas na maioria dos nichos ecológicos, como filoplano e folhas decíduas, insetos mortos e seus habitats, solo, água, poeira de silos e grãos armazenados, mamíferos insetívoros e tecidos humanos com necrose grave (VILAS-BOAS; PERUCA; ARANTES, 2007; MAAGD; BRAVO; CRICKMORE, 2001; PALMA et al., 2014; BRAVO et al., 1998).

Antes que um biopesticida seja comercializado e utilizado no controle de pragas, deve ser testado e garantido a sua biossegurança, de forma a garantir que sua toxicidade não represente risco de dano à saúde humana, ao meio ambiente, ou qualquer efeito adverso frente a organismos não alvo. A utilização de inseticida microbiano é vantajoso, entre outros motivos, por ser específico para um espectro estreito de alvos com atividades que permitem matar somente certas espécies de insetos sem toxicidade contra humanos e outros animais (VILAS-BOAS; PERUCA; ARANTES, 2007; BRAVO et al., 2011; BRAVO et al., 2013; PALMA et al., 2014).

Os produtos à base de Bt têm como alvo insetos pragas incluindo lepidópteros, coleópteros, dípteras, himenópteros, ortópteras, malófagas, além de nematóides, ácaros e protozoários. As larvas dos insetos ingerem os cristais que se dissolvem no ambiente alcalino do intestino liberando proteínas solúveis, as quais formam poros na microvilosidade da membrana apical das células, rompendo as células epiteliais do intestino do inseto alvo,

levando-o a uma severa septicemia e morte. Os produtos à base de Bt podem ser aplicados para pulverização em vários ecossistemas, como folhagem, solos, ambientes aquáticos e grãos armazenados. Além disso, vários genes *cry* também podem ser introduzidos em plantas transgênicas, como tabaco, alfafa, algodão, arroz, maçã, milho, pêra, brócolis, feijão e cana-de-açúcar (SCHNEPF et al., 1998; MONNERAT et al., 2006; BRAVO; GILL; SOBERÓN, 2007; ROH et al., 2007; VILAS-BOAS; PERUCA; ARANTES, 2007; NOGUERA; IBARRA, 2010; BRAVO et al., 2011; YE et al., 2012; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; ADANG; CRICKMORE, 2014; PALMA et al., 2014).

A principal ameaça para o uso de toxinas Cry em plantas transgênicas é o potencial desenvolvimento de insetos resistentes pelo uso contínuo e o fluxo gênico para parentes selvagens. Várias estratégias têm sido propostas para atrasar o desenvolvimento de resistência e impedir o fluxo gênico, como o isolamento de uma nova proteína Cry na natureza, o uso de múltiplas toxinas, refúgios temporais ou espaciais e doses elevadas (SCHNEPF et al., 1998; MONNERAT et al., 2006; VILAS-BOAS; PERUCA; ARANTES, 2007; YE et al., 2012; BRAVO et al., 2013).

As abordagens para conter o aparecimento de resistência de insetos visam melhorar e recuperar a toxicidade. O desenvolvimento de produtos à base de Bt tem o objetivo de ampliar o espectro de pragas de insetos, atingindo novos alvos (SCHNEPF et al., 1998; BRAVO et al., 2011; YE et al., 2012). Para desenvolver novos produtos à base de Bt é necessário o descobrimento de novas proteínas Cry. Nesse sentido, é necessário o desenvolvimento de ferramentas que agilizem a avaliação de genes *cry* de linhagens de Bt. Existem ferramentas baseadas em PCR, com o uso de *primers* específicos e degenerados, mas essas técnicas não respondem bem para a descoberta de novos genes e que normalmente essas ferramentas acabam sendo interessantes para caracterizar genes já conhecidos (ROH et al., 2007; NOGUERA; IBARRA, 2010; ADANG; CRICKMORE, 2014).

As ferramentas de bioinformática são uma estratégia interessante, tendo em vista que agilizam os trabalhos. Os programas de sequenciação identificam a sequência (ou sequências) de codificação de toxinas completas, e o pesquisador completa as etapas de montagem, amplificando por PCR e clonando para a expressão proteica no sistema hospedeiro escolhido (PALMA et al., 2014; YE et al., 2012). As ferramenta de bioinformática disponíveis publicamente ou não preveem genes criados a partir de sequência genômicas, como o caso do método BLAST, que mede a similaridade da sequência, ou não estão sempre disponíveis como é o caso do BtToxin_scanner.

Nesse contexto, esse trabalho consiste em um ambiente que utiliza uma série de programas de bioinformática e um *script* que automatiza a passagem de resultados intermediários, putativos de genes *cry*, entre os estágios de análise, portanto não finalista, para auxiliar o especialista no processo de anotação de regiões codificantes e validação da expressão de genes *cry* em bancada.

1.1 Objetivos

Esse trabalho teve como principal objetivo desenvolver uma *web tool* que auxilie pesquisadores na tarefa de classificar e identificar genes *cry* a partir de sequências de nucleotídeos. Trata-se de um *pipeline* composto por software de bioinformática e que utiliza uma base de dados curada de genes *cry*, descrita na seção 4.2. Para atingir o objetivo geral, foram considerados os seguintes objetivos específicos:

- Fazer a busca na base de perfis e analisar uma sequência de nucleotídeos por meio de modelos ocultos de Markov (HMM, do inglês *Hidden Markov Models*), descrito na seção 3.1;
- Verificar se essa sequência (ou parte dela) forma uma proteína Cry;
- Encontrar e combinar intervalos sobrepostos ou próximos em um único intervalo, listando-o(s) ordenado(s) com base no *E-value* e *score*;
- Agrupar os intervalos e a base de perfis com o intuito de gerar uma matriz de identidade;
- Gerar e apresentar o alinhamento das sequências com maior *score*;
- Implementar um sistema *web* que permita a integração das ferramentas HMMER, BEDTOOLS, BLAST e MUSCLE por meio de *web services*.

Essa ferramenta apresenta uma interface amigável e de fácil navegação que auxilia o pesquisador na busca do conhecimento.

1.2 Justificativa

Existem muitas pragas de insetos que não são suscetíveis às toxinas Cry ou que são controladas de forma ineficiente pelas proteínas Cry identificadas até agora. No caso de plantas transgênicas, por exemplo, uma das ameaças ao controle de pragas é o aparecimento de insetos resistentes, enquanto que no controle de mosquitos vetores de doença, o problema é a baixa atividade do Bti para certas larvas desses mosquitos. Uma das formas de se enfrentar os problemas citados é a busca para isolar novos genes ou linhagens que apresentem um perfil tóxico diferente ou com maior eficiência, com o objetivo de melhorar a toxicidade contra pragas específicas, capazes de atingir novos alvos ou superar o aparecimento de resistência no campo (BRAVO; GILL; SOBERÓN, 2007; NOGUERA; IBARRA, 2010; YE et al., 2012; BRAVO et al., 2013).

Não é uma tarefa fácil encontrar um novo gene *cry* de uma estirpe natural de Bt (ROH et al., 2007; YE et al., 2012). Várias abordagens para procurar novas toxinas com

alta toxicidade e o desenvolvimento de produtos contra um espectro mais amplo de alvos têm sido empregadas com as inovações em biologia molecular:

- reação em cadeia da polimerase (PCR, do inglês *Polymerase Chain Reaction*) para predizer a atividade inseticida de estirpes de Bt já descaracterizadas (BRAVO et al., 1998; ROH et al., 2007; NOGUERA; IBARRA, 2010; YE et al., 2012; PALMA et al., 2014; ADANG; CRICKMORE, 2014);
- construção de bibliotecas de DNA de Bt em *Escherichia coli* (YE et al., 2012; NOGUERA; IBARRA, 2010; PALMA et al., 2014);
- triagem pelo protocolo *Westerm blotting* (YE et al., 2012);
- O desenvolvimento de uma biblioteca de DNA em um mutante acristalífero de Bt, etc (YE et al., 2012; NOGUERA; IBARRA, 2010).

Todos esses métodos citados para encontrar um novo gene apresentam limitações. A estratégia baseada em PCR é limitada a encontrar genes *cry* com alta similaridade para os *primers* usados. Os métodos baseados em biblioteca consomem tempo, são trabalhosos e requerem alto nível de serendipidade. Os outros métodos listados são limitados a encontrarem somente sequências relacionadas a genes *cry* conhecidos (NOGUERA; IBARRA, 2010; ROH et al., 2007; YE et al., 2012).

Nos últimos anos a ênfase mudou na direção de identificar novos genes por meio do sequenciamento, quer da totalidade do genoma da bactéria ou apenas dos megaplasmídeos em que os genes das toxinas Cry são normalmente encontrados. A utilização de ferramentas de bioinformática publicamente disponíveis auxiliam na predição de genes *cry* para sequências genômicas, como é o caso do NCBI Blast, que utiliza comparação por pares de alinhamentos locais para mensurar a similaridade de sequências, ou o *pipeline* computacional BtToxin_scanner, que combina algoritmos de classificação de proteínas para isolar novos genes *cry* (YE et al., 2012; ADANG; CRICKMORE, 2014; PALMA et al., 2014).

As principais ferramentas de bioinformática para o sequenciamento genômico podem identificar genes de toxinas putativas relacionadas a um gene conhecido, porém, podem sofrer com o problema de atualização das suas bases de dados ou não estarem sempre disponíveis para utilização. Outro ponto sensível é a limitação relacionada ao tamanho das sequências que essas ferramentas aceitam, muitas vezes o especialista precisa executar várias ferramentas de bioinformática separadamente para obter uma resposta, ou o resultado é apresentado em uma interface não tão amigável (YE et al., 2012; ADANG; CRICKMORE, 2014; PALMA et al., 2014).

A estratégia desse trabalho visa encontrar resultados putativos, que necessita do especialista para validar, no entanto, as dificuldades citadas serviram de motivação para desenvolver um ambiente *web* acessível, disponível, com desempenho e escalabilidade.

1.3 Organização do trabalho

O trabalho foi dividido em seis capítulos onde são abordadas as considerações gerais da pesquisa, as informações relevantes sobre o *Bacillus thuringiensis*, o detalhamento dos programas de bioinformática utilizados na construção do *pipeline*, bem como o funcionamento da ferramenta, os resultados obtidos por intermédio da sua execução e as considerações para trabalhos futuros.

O Capítulo 2 apresenta um resumo do *Bacillus thuringiensis*, contendo a sua evolução histórica e a importância no controle de pragas agrícolas e mosquitos vetores de doenças, também apresenta características importantes como a sua nomenclatura e caracterização. Esse capítulo ainda trata dos conceitos biológicos relacionados aos genes *cry*, quanto à toxicidade e aos mecanimos de ação no controle biológico de pragas, que os tornam tão importantes. A preocupação com o surgimento de insetos resistentes tem impulsionado os estudos em torno de como diminuir a resistência às toxinas Cry, levando à necessidade de se detectar novos genes *cry* com toxicidade melhorada.

Na busca por detectar novos genes *cry*, o Capítulo 3 apresenta as principais ferramentas de bioinformática relacionadas com esse trabalho. O estudo dessas ferramentas de bioinformática aborda os algoritmos computacionais implementados nos software HMMER, BEDTOOLS, BLAST e MUSCLE.

O Capítulo 4 apresenta o *pipeline* para a descoberta e caracterização de genes *cry* proposto neste trabalho. São descritos cada um dos software utilizados e a implementação do sistema *web*.

O Capítulo 5 apresenta os resultados obtidos pela ferramenta a partir de análises contendo a sequência completa do genoma de *Bacillus thuringiensis* BR145.

O Capítulo 6 apresenta as considerações finais do trabalho e as possíveis direções futuras da pesquisa.

2 *BACILLUS THURINGIENSIS*

O *Bacillus thuringiensis* (Bt) é uma bactéria que forma cristais parasporais compostos de proteínas que são toxinas ativas contra larvas de uma variedade de insetos de diferentes Ordens, permitindo o seu uso em produtos para controle biológico de pragas agrícolas e mosquitos vetores de doenças. A bioatividade específica do Bt ocorre devido a produção desses cristais, que são formados por polipeptídeos conhecidos como proteínas Cry, e apresentam propriedades entomopatogênicas para insetos da Ordem das lepidópteras, dípteras, coleópteras, himenópteras, homópteras, dictiópteras, ortópteras e malófagas, além de nematóides, protozoários e ácaros. No entanto, é seguro para os seres humanos e outros vertebrados, além de ser biodegradável e compatível com práticas agrícolas. A Figura 1 ilustra as Ordens de insetos e as respectivas δ-endotoxinas (ADANG; CRICKMORE, 2014; MONNERAT et al., 2006; ROH et al., 2007). A forma de uma proteína é especificada pela sua sequência de aminoácidos (ALBERTS et al., 2009):

> Existem 20 tipos diferentes de aminoácidos nas proteínas, cada um com propriedades químicas distintas. Uma molécula de proteína é formada a partir de uma longa cadeia de aminoácidos, cada um ligado ao seu vizinho por uma ligação peptídica covalente. As proteínas, são portanto, também chamadas de polipeptídeos. Cada tipo de proteína tem uma sequência exclusiva de aminoácido, e existem milhares de proteínas diferentes, cada qual com a sua própria sequência de aminoácidos (ALBERTS et al., 2009).

Os genes *cry* que codificam proteínas Cry são comumente encontrados em grandes plasmídeos transmissíveis ou mais raramente no cromossomo (PALMA et al., 2014; VILAS-BOAS; PERUCA; ARANTES, 2007; MAAGD; BRAVO; CRICKMORE, 2001; YE et al., 2012). A definição de plasmídeos explica a capacidade desses genes de se multiplicarem mais facilmente dentro das células bacterianas, de maneira independente do cromossomo principal das células (ZAHA; FERREIRA; PASSAGLIA, 2014).

> (...) elementos genéticos extracromossômicos com capacidade de replicação autônoma extracelular (...) é formada por moléculas de DNA de fita dupla circulares, embora existam alguns poucos plasmídeos de DNA de fita dupla lineares, que geralmente ocorrem em bactérias que também possuem cromossomos lineares. (...) prefere-se utilizar o termo transmissível para definir plasmídeos capazes de se manterem num determinado grupo taxonômico de bactérias, mas sem exclusividade (ZAHA; FERREIRA; PASSAGLIA, 2014).

A síntese das proteínas Cry e a formação de esporos são simultâneos, mas a regulação da expressão desses genes podem ser dependentes ou independentes de esporulação. Diferentes combinações de genes *cry* são encontradas em várias cepas de Bt incluindo

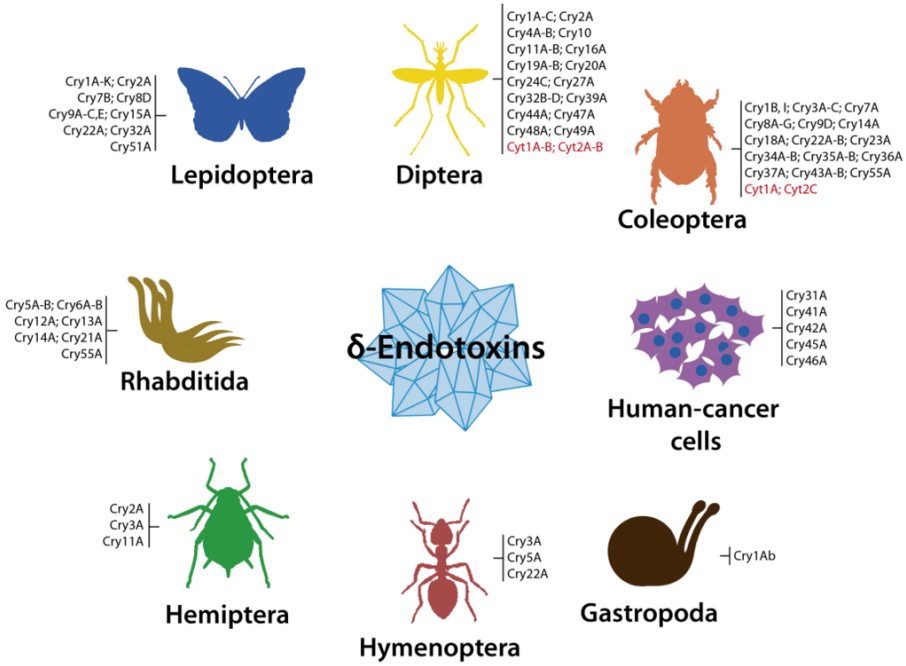


Figura 1 – Resumo dos insetos alvos conhecidos δ-endotoxina de Bt. Fonte: (PALMA et al., 2014).

aquelas com um, dois ou quatro genes diferentes. A transferência desses genes por conjugação é muito frequente em insetos, menos frequente em solos esterilizados e parece não acorrer em solos não esterilizados (VILAS-BOAS; PERUCA; ARANTES, 2007; MAAGD; BRAVO; CRICKMORE, 2001).

O Bt é também uma bactéria chave na construção de plantas transgênicas resistentes a pragas agrícolas. A expressão de determinada toxina Cry em culturas transgênicas tem contribuído para um controle eficiente de pragas de insetos tendo como resultado uma redução significativa do uso de inseticidas sintéticos (SCHNEPF et al., 1998; BRAVO et al., 1998; MONNERAT et al., 2006; ROH et al., 2007; BRAVO et al., 2011; YE et al., 2012; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; PALMA et al., 2014).

Desde que se tornou alvo de estudos até hoje, o Bt é reconhecido pela grande variabilidade de linhagens, seja pelo número enorme de estirpes isoladas ao redor do mundo, pela quantidade de sorotipos conhecidos, o grande número de diferentes sequências de genes *cry* acumuladas até agora ou o número de ferramentas de caraterização molecular que têm sido desenvolvidas para estudo (ADANG; CRICKMORE, 2014; MAAGD; BRAVO; CRICKMORE, 2001; NOGUERA; IBARRA, 2010; ROH et al., 2007), (seção 2.5).

2.1 Evolução Histórica

A bactéria Bt é o patógeno de inseto de maior sucesso usado no controle de insetos, representando aproximadamente 2% do mercado mundial de inseticidas, o que equivale a

cerca de 80% de todos os biopesticidas vendidos. O uso de produtos Bt na agricultura e culturas florestais remonta há mais de 70 anos, quando se tornou disponível na França. Nas últimas décadas mais de 700 sequências de genes de Bt que codificam proteínas de cristal (Cry) ou δ-endotoxinas foram identificadas (VILAS-BOAS; PERUCA; ARANTES, 2007; ROH et al., 2007; NOGUERA; IBARRA, 2010; BRAVO et al., 2011; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; PALMA et al., 2014).

O Bt foi descrito primeiro por Ernst Berliner quando este isolou uma espécie na província de Thuringia na Alemanha em 1911, por esse motivo o nome *Bacillus thuringiensis*. Antes, porém, em 1901, um biólogo japonês, Shigetane Ishiwatari, descobriu uma bactéria como agente causador de uma doença que afetava o bicho da seda e que ainda não havia sido descrita. Thomas A. Angus, em 1956, demonstrou que inclusões de proteínas cristalinas formadas durante a esporulação foram responsáveis pela ação inseticida de Bt (BRAVO et al., 2013; ROH et al., 2007; PINTO et al., 2003).

Em 1977 foi descoberta uma subespécie de Bt, chamada *israelenses*, que é um dos mais efetivos e potentes pesticidas biológicos contra mosquitos e borrachudos, promovendo benefícios ambientais e medicinais. O surgimento do Bti aconteceu em um momento de resistência de mosquitos e borrachudos aos pesticidas sintéticos. O Bti é um exemplo de sucesso em cooperação entre indústria e organizações governamentais, como a OMS (Organização Mundial da Saúde). Além disso, o seu custo de desenvolvimento é estimado em 1/40 em relação a um novo pesticida químico sintético (VILAS-BOAS; PERUCA; ARANTES, 2007; SCHNEPF et al., 1998; ROH et al., 2007).

Os pesticidas baseados em proteínas Cry são os agentes de controle biológico de pragas mais amplamente utilizados e, geralmente, têm um baixo custo de desenvolvimento e registro. Em 1995 as vendas projetadas de Bt foram de US$90 milhões, com 182 produtos baseados em Bt registrados pela Agência de Proteção Ambiental (EPA) nos EUA, mostrando ser o agente de controle de pragas biológico mais utilizado no mercado de inseticidas. Já em 1998 foram registrados nos EUA mais de 200 produtos de Bt (SCHNEPF et al., 1998).

Na década de 1980, os cientistas revelaram que as plantas poderiam ser geneticamente modificadas. Em 1987 inseticidas de genes *cry* foram introduzidos e expressados em tecidos de tabaco e tomates, mas só a partir de 1996 começou a produção de proteínas Cry em plantas, tornando-se um marco na biotecnologia agrícola. Variedades de batata, algodão e milho modificados contendo o gene *cry* foram vendidas aos produtores. O desenvolvimento de culturas Bt que expressam o gene *cry* resulta no cultivo resistente ao ataque de pragas agrícolas (SCHNEPF et al., 1998; MAAGD; BRAVO; CRICKMORE, 2001; ROH et al., 2007).

Bons exemplos do cultivo de transgênicos vêm da China, EUA e Índia. Na China, em 1999, o cultivo de algodão-Bt reduziu de 22% para 4,7% a aplicação de pesticidas

nos campos. Os EUA, em 2003, produziram 2,4 milhões de toneladas extra de comida e fibra, aumentando a renda agrícola em US$1,9 bilhões com o cultivo de transgênicos de canola, milho, algodão, papaia, abóbora e soja. Além disso, o cultivo de transgênicos também reduziu o uso de pesticidas em 21 mil toneladas. O algodão-Bt na Índia reduziu a aplicação de inseticida em 70%. Além de uma economia de US$30 por hectare, representou um aumento de 80 a 87% nos campos de algodão. Uma das vantagens no cultivo de plantas transgênicas é a continuidade da produção da proteína Cry dentro das células, protegendo-a da inativação causada pelos raios UV (ROH et al., 2007; BRAVO et al., 2011; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013).

Em 2009 foram cultivados mais de 40 milhões de hectares em escala mundial com culturas Bt de milho ou algodão. Em 2010 esse número aumentou para mais de 58 milhões de hectares. Isso representa uma significativa redução no uso de inseticidas sintéticos e a supressão de certas pragas de insetos, além da redução dos custos na gestão do campo. As culturas Bt mais importantes são soja, milho algodão e canola (BRAVO et al., 2011; BRAVO et al., 2013; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013).

A produção de milho transgênico no Brasil está aumentando. O melhoramento genético, com a introdução de genes que codificam a proteína Cry, tem possibilitado a redução de até 90% da densidade populacional de pragas, permitindo maiores desenvolvimento e rentabilidade. A produção de milho no Brasil é uma atividade lucrativa e importante para a balança comercial, onde todos os estados brasileiros possuem uma parcela da produção de milho no montante nacional. Segundo o último censo agropecuário, o estado do Paraná apresentou os resultados mais expressivos, com mais de 9 milhões de toneladas produzidas (CUSTODIO et al., 2016).

A Organização para Cooperação e Desenvolvimento Econômico (OECD) estima que até 2020 os bioinseticidas representem 20% do mercado mundial de inseticidas (ROH et al., 2007).

2.2 Características Gerais

A classificação de um gene *cry* diz respeito à similaridade de aminoácidos de uma proteína Cry com outras proteínas Cry. Já foram identificadas mais de 700 sequências de genes de Bt, a única bactéria que produz esse tipo de proteína. As sequências das principais proteínas que conferem toxicidade a insetos estão divididas em três famílias de proteínas não relacionadas filogeneticamente que podem ter diferentes modos de ação (VILAS-BOAS; PERUCA; ARANTES, 2007; BRAVO et al., 2011; YE et al., 2012; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; BRAVO et al., 2013; ADANG; CRICKMORE, 2014):

- toxinas de cristal (Cry);

- toxinas da família citolítica (Cyt);
- toxinas inseticida Vip.

Destes grupos, o maior é o da família de proteínas Cry, com mais de 70 categorias diferentes com toxicidade reconhecida a insetos lepidópteros, coleópteros, entre outros. As toxinas pertencentes a cada grupo Cry compartilham menos de 40% dos aminoácidos identificados com proteínas de outros grupos. Os tamanhos dessas proteínas variam de 369 a 1344 aminoácidos (ADANG; CRICKMORE, 2014; BRAVO et al., 2013; BRAVO et al., 2011).

Nas últimas décadas foram identificados e localizados em grandes plasmídeos uma variedade de genes de Bt que codificam proteínas inseticidas, assim descritos (YE et al., 2012; PALMA et al., 2014; CONSTANSKI et al., 2015):

- 743 genes *cry* divididos em 73 categorias primárias;
- 96 genes *vip* em quatro categorias primárias;
- 34 genes *cyt* divididos em três categorias primárias.

Muitas dessas proteínas têm propriedades pesticidas úteis para o controle de pragas de insetos na agricultura. Além da vantagem de não ter nenhum alvo vertebrado conhecido. Proteínas de Bt são altamente específicas para seus hospedeiros e ganharam importância mundial como uma alternativa aos inseticidas sintéticos (PALMA et al., 2014; MAAGD; BRAVO; CRICKMORE, 2001).

2.2.1 Nomenclatura

A identificação e clonagem do primeiro gene de proteína de cristal inseticida de Bt aconteceu em 1981. O Comitê de Nomenclatura de Toxinas de *Bacillus thuringiensis* nomeia uma nova toxina dependendo do grau de identidade de aminoácidos alinhados, ou seja, são classificadas com base em sequências homólogas de aminoácidos. É importante salientar que esse critério não implica em uma estrutura de proteína similar, nem classe de hospedeiros ou modo de ação. Como a nomeação das toxinas não leva em conta a sua toxicidade, existem toxinas que são ativas contra a mesma Ordem de inseto, porém, não necessariamente compartilham qualquer similaridade em seus nomes (MAAGD; BRAVO; CRICKMORE, 2001; ROH et al., 2007; PALMA et al., 2014; ADANG; CRICKMORE, 2014).

O nome de uma nova proteína é composto de cinco partes, sendo que cada parte representa uma classe hierárquica. O nome da proteína consiste no mnemônico Cry, Cyt ou Vip e quatro classes hierárquicas. A primeira classe é representada por números arábicos,

a segunda classe utilizando letra maiúscula, a terceira classe usando letra minúscula e a quarta classe empregando números arábicos (MAAGD; BRAVO; CRICKMORE, 2001; PALMA et al., 2014), como ilustra a Figura 2.

Na classificação primária as proteínas compartilham menos de 45% de identidade na sequência. Na classificação secundária esse compartilhamento é menor que 78% de identidade. Na classificação terciária a identidade das sequências alinhadas é menor que 95%. Por fim, a classificação quaternária é associada a proteínas que compartilham mais de 95% de identidade. Embora algumas proteínas possam ter diferentes classificações quaternárias, elas podem compartilhar sequências idênticas de aminoácidos (PALMA et al., 2014; MAAGD; BRAVO; CRICKMORE, 2001; ROH et al., 2007; ADANG; CRICKMORE, 2014). Na Figura 2 é apresentada a classificação descrita acima.

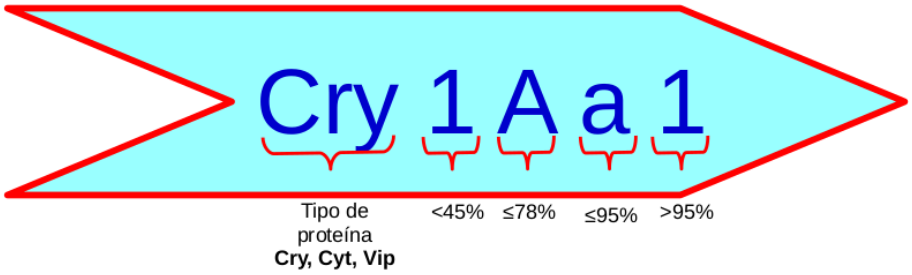


Figura 2 – Sistema de nomenclatura utilizado para classificação das δ-endotoxinas (Cry e Cyt) e toxinas secretadas (Vip).

2.3 Mecanismo de Ação

As proteínas de cristais no Bt podem ter várias formas dependendo de suas composições de protoxina: bipiramidal (Cry1), cuboidal (Cry2), retangular plano (Cry3A), irregular (Cry3B), esférico (Cry4A e Cry4B) e romboidal (Cry11A). Na Figura 3 é possível ver uma microscopia eletrônica que apresenta algumas conformações de cristais padrão de Bt.

A bioatividade específica do Bt devido à produção de proteínas cristais, codificadas pelos genes *cry*, pertencem a uma classe de toxinas bacterianas formadoras de poro (PFT, do inglês *pore-forming toxins*) e são considerados agentes patógenos oportunistas para larvas de insetos (BRAVO et al., 2011; SCHNEPF et al., 1998; PINTO et al., 2003).

A atividade inseticida da proteína Cry, ou δ-endotoxinas, apresenta toxicidade contra várias Ordens de insetos, o que levou ao desenvolvimento global de bioinseticidas para controle de pragas, e podem ser aplicados a vários ecossistemas como, por exemplo, folhagem, solo, ambientes aquáticos e grãos armazenados. Além disso, genes *cry* podem ser introduzidos em microorganismos, e também podem ser inseridos em plantas (VILAS-BOAS; PERUCA; ARANTES, 2007; BRAVO et al., 2011; BRAVO et al., 1998).

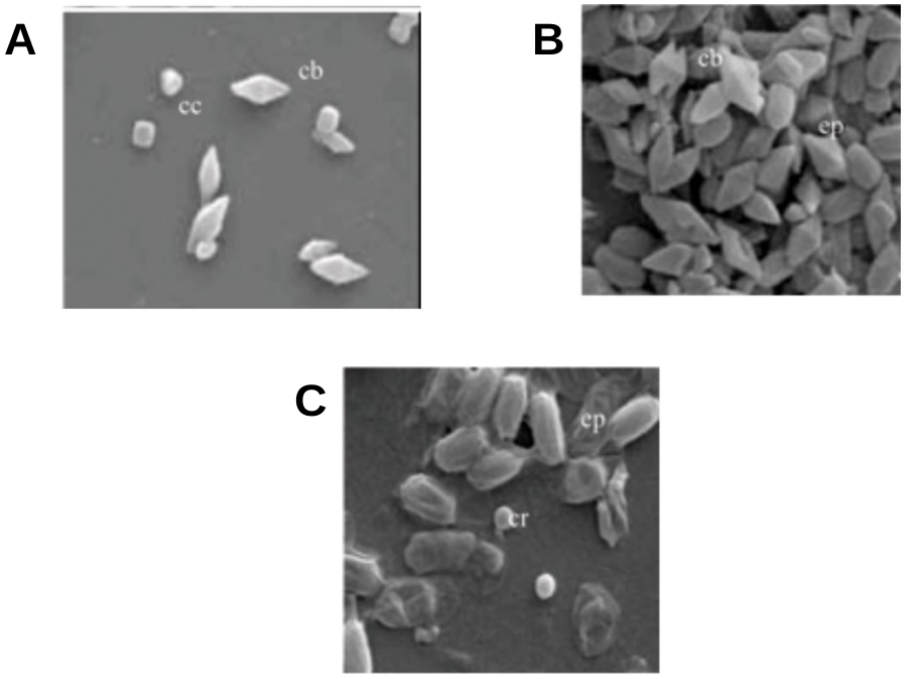


Figura 3 – Microcospia eletrônica de varredura do complexo espro-cristal: (A) cb: cristal bipiramidal, cc: cristal cubóide; (B) cb: cristal bipiramidal, ep: esporo; (C) cr: cristal redondo, ep: esporo. Fonte: (ABREU, 2006)

A Figura 4 ilustra os detalhes do modo de ação da toxina Cry, que consiste em um processo multipasso que envolve a interação com várias moléculas receptoras levando à inserção na membrana e ao rompimento da célula. A ação primária das toxinas Cry exerce efeito patológico ao romper as células epiteliais no intestino do inseto alvo formando poros líticos na microvilosidade da membrana apical das células, levando-o a uma severa septicemia e morte. Os cristais são ingeridos pelas larvas de insetos sensíveis e dissolvem-se no ambiente alcalino do intestino, liberando proteínas solúveis (BRAVO; GILL; SOBERÓN, 2007; MAAGD; BRAVO; CRICKMORE, 2001; ROH et al., 2007; ADANG; CRICKMORE, 2014).

2.3.1 Processo de Intoxicação por Toxinas Cry

A especificidade do inseto é amplamente determinada pela ligação das toxinas Cry a microvilosidade da superfície das células do intestino médio da larva (BRAVO et al., 2011). Os cristais são compostos de protoxinas que para se tornarem ativas é necessário a ingestão pela larva do inseto suscetível. A ação do Bt depende da toxina inseticida que está ativa durante o processo patogênico, produzindo uma variedade de fatores de virulência que contribuem para a morte do inseto. Depois da solubilização as protoxinas são processadas por proteases no intestino do inseto para torná-las ativas.

As toxinas Cry1A, como exemplo de atividade patogênica, se ligam às proteínas caderina de pelo menos 6 espécies de lepidópteras, *Manduca sexta*, bicho-da-seda, lagarta

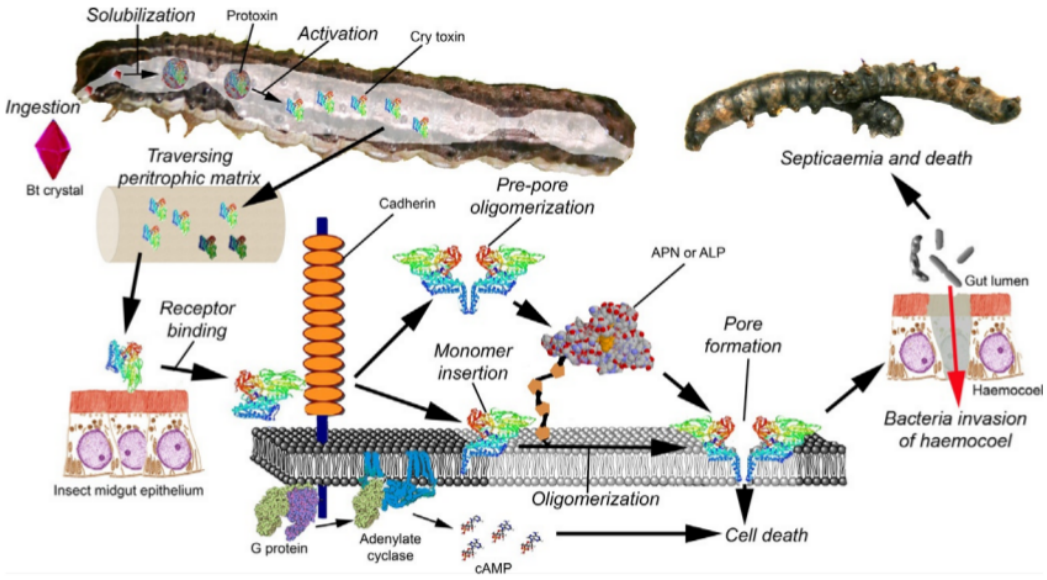


Figura 4 – Representação do modo de ação da toxina Cry nas células epiteliais do intestino do inseto. Fonte: (ADANG; CRICKMORE, 2014)

das maçãs, *Helicoverpa armigera*, lagarta rosada e *Ostrina nubilalis* (BRAVO et al., 2011).

Já nos mosquitos - importantes vetores de doenças humanas como dengue, febre amarela e malária, entre outras - algumas das toxinas Cry que apresentam toxicidade são: Cry1, Cry2, Cry4, Cry11 e Cry9. Uma variedade particular de Bt, *israelenses*, tem sido usada mundialmente para o controle desses vetores, apresentando alta toxicidade ao *Aedes aegypti*, produzindo a inclusão de cristais formados principalmente por Cry4Aa, Cry4Ba e Cry11Aa (BRAVO et al., 2011).

As proteínas de ligação Cry3Aa e Cry3Bb ao se ligarem a uma proteína caderina apresentam toxicidade melhorada contra diversas larvas de insetos coleópteros. Enquanto uma proteína ALP que se liga a toxina Cry1B é um receptor Cry putativo contra uma larva específica de insetos coleópteros. Isso significa que apesar de serem proteínas diferentes e agirem sobre insetos diferentes, sendo ou não do mesmo grupo, apresentam mecanismos semelhantes, sendo portanto conservados em diferentes Ordens de insetos (BRAVO et al., 2011).

As toxinas Cry têm um espectro definido de atividade inseticida restrita a poucas espécies em uma Ordem particular. São poucas as toxinas que tem um espectro de atividade que alcancem duas ou três Ordens de insetos (ROH et al., 2007).

2.3.2 Solubilização

As protoxinas são solubilizadas, ou seja, dissolvidas sob condições alcalinas do intestino do inseto (um pH superior a 7) e digerida pelas enzimas digestivas tripsina ou quimotripsina, gerando um fragmento tóxico de aproximadamente 60 a 70 kDa. As

protoxinas têm massa entre 130 e 140 kDa. Essas proteases removem a metade da molécula da extensão C-terminal bem como um pequeno fragmento N-terminal das proteínas Cry (SCHNEPF et al., 1998; MAAGD; BRAVO; CRICKMORE, 2001; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; ROH et al., 2007).

As toxinas Cry têm a capacidade de inserir-se nas membranas e formar poros sob certas condições, que inclui alta concentração da toxina, tempo de incubação e pH relativamente baixo (SCHNEPF et al., 1998; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; ROH et al., 2007). Abaixo lista-se as condições específicas para a formação de cada tipo de poro:

- em condições alcalinas do intestino (pH 9,5) forma canais de cátions de diferentes tamanhos;
- em condições ácidas do intestino (pH 6,0) forma canais de ânions de diferentes tamanhos.

A eventual lise celular osmótica se dá pela formação de canais de cátions, ânions e o afluxo de água que resulta em inchaço celular. No entanto, grandes diferenças na fisiologia do intestino das diferentes Ordens de insetos podem levar a mudanças na especificidade. No caso das lepidópteras e dípteras a serina é a principal protease no suco intestinal. Já no caso dos coleópteros são, principalmente, as proteases cisteína e aspártica (MAAGD; BRAVO; CRICKMORE, 2001; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; ADANG; CRICKMORE, 2014).

2.3.3 Função do Receptor e Proteínas de Ligação Cry

A interação da toxina Cry com diferentes proteínas presentes nas células do intestino médio da lepidóptera, por exemplo, é um processo complexo envolvendo múltiplas proteínas identificadas como receptoras: caderina (CAD), aminopeptidase N (APN) e fosfatase alcalina (ALP). Outros componentes têm a capacidade de interagir com as toxinas Cry-3D, tal como glicolipídios ou proteínas intramoleculares (V-ATPase). As aminopeptidases e as caderinas interagem de modo consecutivo com diferentes estruturas da toxina (PARDO-LOPEZ; SOBERÓN; BRAVO, 2013; ROH et al., 2007; MAGALHÃES, 2006).

A identificação de moléculas receptoras e epitopos (antígenos) de ligação ajudam no desenvolvimento de estratégias para lidar com o potencial problema de insetos resistentes às toxinas Cry (BRAVO; GILL; SOBERÓN, 2007) (seção 2.4). A seguir é feito uma breve descrição das proteínas identificadas como receptoras.

Aminopeptidase

Aminopeptidase N (APN) foi uma das primeiras proteínas identificadas como receptores Cry putativos em insetos. As APNs estão ligadas à borda em escova do intestino de onde elas clivam os aminoácidos do N-terminal de peptídeos. É um elemento importante no processo de formação de poros das células epiteliais (ADANG; CRICKMORE, 2014; MAGALHÃES, 2006).

Análises filogenéticas de APNs em lepidópteras agrupam essas proteínas em sete classes. Os membros da classe APN1 são os mais altamente expressos no tecido do intestino médio. Diversos APNs têm sido identificados como receptores putativos de Cry11Aa e Cry11Ba, com alta afinidade no *Aedes aegypti*, causador da febre amarela, dengue, entre outras; e Cry11Ba no *Anopheles albimanus* e *An. gambiae*, mosquitos da malária (ROH et al., 2007; ADANG; CRICKMORE, 2014).

Caderina

As proteínas da família das caderinas (CAD) são receptoras de toxinas Cry funcionais. As proteínas CAD se ligam às toxinas Cry localizadas na maioria das vezes na membrana da borda em escova das células do intestino médio. As caderinas também estão presentes na ancoragem da membrana basal das células epiteliais do intestino das larvas da *Manduca sexta* e *Lymantria dispar* (BRAVO et al., 2011; ADANG; CRICKMORE, 2014).

Fosfatase

A fosfatase (ALP) é uma enzima que catalisa a hidrólise de fosfatos orgânicos em um ambiente ácido ou alcalino. As ALPs do intestino médio ligadas à membrana consistem no maior grupo de proteínas de ligação Cry identificadas em larvas de lepidóptera, coleóptera e díptera (ADANG; CRICKMORE, 2014).

2.4 Resistência às Toxinas Cry

Há várias espécies de insetos que tornam-se resistentes a um ou múltiplos inseticidas sintéticos. Também, a partir dos anos 1980, um número de populações de várias espécies diferentes com diferentes níveis de resistência às proteínas de Bt foram detectados. Em menos de duas décadas de uso intensivo das subespécies *kurstaki* e *aizawai*, foram isolados insetos resistentes em numerosas regiões geograficamente distantes do mundo. O uso imprudente de toxinas Cry pode torná-las ineficazes contra importantes pragas agrícolas (SCHNEPF et al., 1998; ROH et al., 2007).

A principal desvantagem de plantas-Bt é o potencial desenvolvimento de insetos resistente às toxinas Cry e o fluxo gênico para parentes selvagens devido ao uso contínuo de

produtos Bt (VILAS-BOAS; PERUCA; ARANTES, 2007; YE et al., 2012; MONNERAT et al., 2006).

Várias estratégias, como o uso de múltiplas toxinas, a criação de refúgios temporais ou espaciais e doses elevadas das toxinas, têm sido propostas para atrasar o desenvolvimento de resistência e impedir o fluxo gênico. A procura por novas toxinas com alta toxicidade é uma das maiores abordagens nesse contexto (BRAVO et al., 1998; VILAS-BOAS; PERUCA; ARANTES, 2007; NOGUERA; IBARRA, 2010; YE et al., 2012).

2.4.1 Produtos a Base de Bt para o Controle de Insetos

Bt produz grandes quantidades de proteínas de cristal, tornando-o um candidato para o desenvolvimento de biopesticidas Cry melhorados. Foram desenvolvidos diferentes produtos Bt para o controle de insetos na agricultura e contra espécies de mosquitos. Esses produtos são baseados na preparação de esporos de cristais (BRAVO et al., 2011; SCHNEPF et al., 1998). Os produtos a base de Bt podem ser usados de duas maneiras: na pulverização e no desenvolvimento de plantas transgênicas.

Um avanço importante na redução de inseticidas sintéticos na agricultura aconteceu com o desenvolvimento de culturas transgênicas capazes de expressarem as toxinas Cry de Bt (BRAVO et al., 2011).

Os produtos baseados em Bt var. *kurstaki* (Btk) são efetivos no controle de muitas lepidópteras, que são pragas em culturas importantes, que se alimentam de folhas, pois são desfolhadoras, e de frutos também, como a lagarta da goiaba, da maçã, entre outras. O (Btk) HD-1 expressa as proteínas Cry1Aa, Cry1Ab, Cry1Ac e Cry2Aa, enquanto o (Btk) HD-73 produz Cry1Ac. Os produtos baseados em Bt var. *aizawai* são especialmente ativos contra larvas de lepidópteras que se alimentam de produtos armazenados. O produto HD-137, baseado em Bt var. *aizawai*, produz as toxinas Cry1Aa, Cry1Ba, Cry1Ca e Cry1Da. Já os produtos baseados em Bt *san diego* e Bt *tenebrionis* são ativos contra pragas de besouros na agricultura, produzindo a toxina Cry3Aa. Produtos baseados em Bti são usados no controle de mosquitos que são vetores de doenças em humanos, como a dengue e a malária. Produtos Bti contém as toxinas Cry4A, Cry4B, Cry11A e Cyt1Aa, entre outras (BRAVO et al., 2011). Certas combinações de proteínas Cry apresentam efeitos sinérgicos sobre o organismo alvo, com toxicidade melhorada do que se fossem aplicadas isoladamente.

Produtos pulverizáveis baseados em Bt, apesar da sua eficiência, têm algumas limitações na agricultura. É possível notar o declínio muito rápido tanto na quantidade de esporos quanto na toxicidade. As toxinas Cry expostas à radiação do sol mostram-se sensíveis e acabam tendo uma atividade limitada contra as larvas de insetos, configurando-se em uma desvantagem (SCHNEPF et al., 1998; BRAVO et al., 2011).

Sustentabilidade do Uso de Bt

O uso de *spray* pulverizadores de Bt como um inseticida tem várias desvantagens (ROH et al., 2007; ADANG; CRICKMORE, 2014):

- não pode ser aplicado uniformemente em todas as partes da planta;
- não atinge as pragas que estão no interior dos tecidos das plantas;
- é suscetível à rápida degradação se exposto à luz UV;
- é possível de ser removido por escoamento da água.

Os cultivos transgênicos com genes *cry* de Bt podem superar essas desvantagens. Esses cultivos eliminam as dificuldades na segmentação de pragas que escavam os tecidos vegetais, além de diminuir o trabalho e o custo associado à aplicação de *sprays*. Entre as vantagens, podemos citar (ROH et al., 2007):

- não tem efeitos nocivos em vertebrados e humanos ou ao ambiente ecológico;
- tem baixo impacto em organismos não alvo;
- possui um espectro estreito, permitindo-lhe matar somente certas espécies de insetos, principalmente as lepidópteras que se alimentam de folhas.

Para o uso sustentável de produtos à base de Bt é necessário que haja (NOGUERA; IBARRA, 2010; BRAVO et al., 2011; YE et al., 2012; ADANG; CRICKMORE, 2014):

- coleções de Bt isolados, proteínas de cristal e estirpes de espécies relacionadas;
- investigações relacionadas à persistência de proteínas de cristal no meio ambiente e possível efeito a longo prazo;
- desenvolvimento de estratégias de gerenciamento do aumento de resistência;
- engenharia genética sobre os genes de Bt para melhorar os cultivos transgênicos.

2.4.2 Esforços para Aumentar a Toxicidade

Vários esforços têm sido empregados para melhorar e ampliar a atividade para novos produtos bioinseticidas.

Desde 1989, vários grupos independentes aplicaram a tecnologia de eletroporação para transformar células vegetativas com DNA de plasmídeo (SCHNEPF et al., 1998). A eletroporação é um método que consiste na aplicação de um pulso de alta voltagem,

induzindo poros reversíveis nas membranas celulares, para introduzir macromoléculas em células vegetais (MONQUERO, 2005).

A tecnologia de inserção de *replicons* de plasmídeos de Bt também é usada para inserir genes *cry* clonados em Bt. Além da utilização de vetores de integração para inserir genes *cry* por recombinação homóloga em plasmídeos ou cromossomos (SCHNEPF et al., 1998).

Os sistemas de recombinação têm sido desenvolvidos para construir cepas de Bt recombinante para novos produtos bioinseticidas (SCHNEPF et al., 1998).

Uma das vertentes dos projetos envolvendo o Bt é a transferência das proteínas Cry para outras bactérias que não produzem o endosporo. Uma delas é a *Pseudomas*.

A produção de *Pseudomas* recombinante tem sido utilizada para formular o produto a base de biopesticidas aquoso concentrado, que consiste na inclusão encapsulada em células mortas. As toxinas Cry permitem o desenvolvimento dessa bactéria em larvas de insetos mortos ou debilitados. Essa técnica mostrou o aumento da persistência no ambiente, além de que se combinadas, podem expandir a gama de insetos alvos a serem controlados. O objetivo dessa técnica é prolongar a persistência de proteínas Cry no campo, visto que os cristais são sensíveis a luz solar (SCHNEPF et al., 1998).

Vários genes *cry* foram inseridos em plantas. Os biopesticidas baseados em sementes melhoradas são menos prejudiciais ao meio ambiente do que inseticidas sintéticos e, normalmente, não afetam insetos benéficos, como os predadores naturais e parasitas. Outra vantagem é a possibilidade de expandir a variedade de pragas alvo, incluindo insetos de sucção, insetos que vivem na raiz e nematoides (SCHNEPF et al., 1998; ROH et al., 2007; VILAS-BOAS; PERUCA; ARANTES, 2007; BRAVO et al., 2011; PARDO-LOPEZ; SOBERÓN; BRAVO, 2013).

2.4.3 Estratégias de Gerenciamento de Resistência

As estratégias para o gerenciamento de resistência dependem fortemente de pressupostos teóricos e modelos computacionais de simulação de crescimento de população de insetos sob várias condições. O objetivo é tentar prevenir ou diminuir a seleção dos indivíduos raros portadores de genes resistentes e conseguir manter a frequência dos genes resistentes suficientemente baixa para o controle de insetos.

As estratégias propostas incluem a utilização de múltiplas toxinas, rotação de culturas, alta ou ultra dosagens, refúgio espacial ou temporal. Cada praga de cultura requer uma implementação específica de determinada estratégia, que pode incluir o uso de pulverização e culturas transgênicas de Bt.

Os predadores e parasistas são inimigos naturais das larvas de insetos alvos, e

também podem ajudar a retardar o desenvolvimento de resistência ao Bt. Eles podem influenciar o desenvolvimento de resistência, ao preferirem insetos intoxicados, suscetíveis ou saudáveis.

Projetar uma estratégia de gerenciamento de resistência requer a participação de todos os profissionais envolvidos no cultivo: os fornecedores da tecnologia, as companhias de sementes, os consultores de culturas, órgãos reguladores, e, principalmente, os produtores. Foi o que começou a acontecer a partir de 1988, quando companhias de desenvolvimento de biopesticidas de Bt formaram um Grupo de Trabalho de Gerenciamento de Bt para promover a pesquisa do uso de produtos baseados em Bt. O objetivo é a melhor compreensão da ação combinada entre as toxinas Cry, as bactérias hospedeiras, os organismos alvo e os ecossistemas que eles compartilham, permitindo o efetivo uso de toxinas Cry para o gerenciamento de pragas (SCHNEPF et al., 1998; VILAS-BOAS; PERUCA; ARANTES, 2007; BRAVO et al., 2011).

Rotação de Plantas e Pulverização

A rotação de plantas e a pulverização são duas estratégias para conter insetos resistentes. No caso da pulverização utiliza-se uma toxina de Bt particular com outros tipos de toxina que se ligam a diferentes receptores. A utilização de pulverização combinada com outras toxinas de Bt seria uma desvantagem para os insetos resistentes a um tipo específico de toxina Cry durante a próxima temporada de crescimento, resultando em uma diminuição da frequência do correspondente gene resistente. A reversão a suscetibilidade para esse tipo de toxina Cry deveria ocorrer dentro da temporada de crescimento (SCHNEPF et al., 1998).

Combinar Altas Doses e Refúgio

Uma tática de gerenciamento à resistência é a combinação de uma estratégia de alta dose com o uso de refúgio. O refúgio consiste em áreas livres de toxina. A estratégia consiste em expressar doses de toxinas Cry que quase todos os portadores heterozigotos de alelos resistentes seriam mortos. A probabilidade dos sobreviventes se acasalarem com os insetos sensíveis abrigados nas proximidades do refúgio é maior. Consequentemente, a população de insetos resistentes homozigotos seria significativamente reduzida (SCHNEPF et al., 1998; VILAS-BOAS; PERUCA; ARANTES, 2007; BRAVO et al., 2011; NOGUERA; IBARRA, 2010).

Outra estratégia é fornecer refúgios pré-embalados. Uma plantação específica com o uso de sementes misturadas de plantas que expressam a toxina e livres de toxina. Essa estratégia, no entanto, só é efetiva para espécies de insetos cujas larvas se movimentam muito pouco entre as plantas ou que os insetos adultos conquistem seus companheiros visualmente a curta distância (BRAVO et al., 2013; SCHNEPF et al., 1998).

Expressão de Múltiplas Proteínas Cry

A expressão de múltiplas proteínas Cry em culturas ou incorporar múltiplas proteínas em pulverização de Bt combinado com o uso de refúgio têm modos de ação diferentes com respeito ao mecanismo de resistência do inseto. Essa estratégia poderia ser implantada à toxinas Cry que reconhecem diferentes receptores na mesma espécie alvo (SCHNEPF et al., 1998).

A genética de populações nos mostra que é mais provável que os genes de resistência sejam heterozigotos na população. Com a frequência da resistência baixa, existe uma maior probabilidade de cruzamento entre um indivíduo heterozigoto para resistência com um homozigoto suscetível (CARNEIRO et al., 2009).

Essa estratégia combina toxinas Cry de Bt com outras proteínas inseticidas (SCHNEPF et al., 1998). Então, a dose para matar o heterozigoto é maior do que a dose para o homozigoto suscetível e, também, menor do que a do homozigoto resistente (CARNEIRO et al., 2009).

Assim, encontrar um inseto homozigoto resistente para um gene é raro, encontrar insetos homozigotos resistentes para múltiplos genes é extremamente raro. Então, o desenvolvimento dessa estratégia em cultivos transgênicos ou pulverização controlaria insetos homozigotos para um ou dois genes resistentes e heterozigotos para outro gene. O acasalamento de indivíduos resistentes com suscetíveis produzem indivíduos suscetíveis devido à característica recessiva dos alelos resistentes (SCHNEPF et al., 1998; BRAVO et al., 2011).

Uma condição essencial para o sucesso dessa estratégia é que cada um dos inseticidas seja altamente eficaz para os homozigotos suscetíveis. A avaliação de sucesso depende de algumas variáveis, tais como o ambiente, o inseto e o Bt que será expresso na planta, ou o inseticida (SCHNEPF et al., 1998; CARNEIRO et al., 2009).

2.5 Detecção de Novos Genes *cry*

Encontrar um novo gene *cry* de uma estirpe natural de Bt e que tenha um bom potencial inseticida não é um processo fácil, porque uma estirpe típica de Bt abriga entre um e seis genes *cry*. A descoberta de novos genes de toxinas tem crescido com as inovações em biologia molecular, mais especificamente por PCR e sequenciamento de nova geração (ROH et al., 2007; ADANG; CRICKMORE, 2014).

Originalmente, a decoberta de novos genes começou com a categorização de toxinas. Em seguida a ênfase passou a ser a identificação de estirpes com atividades úteis. Por último, passou-se para a identificação de novos genes por meio de sequenciamento da totalidade do genoma da bactéria, ou apenas os mega-plasmídeos nos quais os genes da

toxina são normalmente encontrados (ADANG; CRICKMORE, 2014).

Uma alternativa para o isolamento de uma nova proteína de toxina Cry na natureza é a evolução genética *in vitro*, com o objetivo de melhorar a toxicidade contra pragas específicas, ampliar o espectro de pragas de insetos e recuperar a toxicidade no caso do aparecimento de resistência evoluída no campo (BRAVO et al., 2013; YE et al., 2012).

A amplificação de DNA por PCR para predizer a atividade inseticida de estirpes de Bt requer o *primer* do tamanho inteiro da sequência, de aproximadamente 3,6kb, contendo a região codificante completa no gene alvo. Esse método além de demorado é trabalhoso e limitado a encontrar genes *cry* com alta sequência de similaridade para os *primers* usados. Poucos deles são relatados como capazes de identificar um novo gene *cry*. Outra deficiência é não conseguir obter o tamanho completo da sequência do gene, além de não distinguir entre genes expressados e silenciados (YE et al., 2012; NOGUERA; IBARRA, 2010; PALMA et al., 2014; BRAVO et al., 1998; ROH et al., 2007).

Outros métodos que também são utilizados para a identificação de genes *cry* consistem na construção de bibliotecas de DNA de Bt em *Escherichia coli*, a triagem pelo protocolo *Westerm blotting*, o método baseado em hibridização, e o desenvolvimento de uma biblioteca de DNA em um mutante acristalífero de Bt. Esses métodos baseados em biblioteca, no entanto, consomem tempo, são trabalhosos e requerem altos níveis de serendipidade (NOGUERA; IBARRA, 2010; ROH et al., 2007; YE et al., 2012).

A tecnologia NGS (do inglês, *next-generation sequencing*) está disponível para a descoberta de genes de toxinas inseticidas antes desconhecidas. Fornece estrutura para obter as sequências genômicas completas. A tecnologia NGS tem um custo médio muito maior que a estratégia baseada em PCR para identificar genes *cry*, embora apresente uma significativa relação custo-benefício (YE et al., 2012; PALMA et al., 2014).

Existem ferramentas de montagem *de novo* usando software de bioinformática, software comerciais (*CLC Genomics Workbench*) ou montadores *opensource* na distribuição BioLinux. Várias ferramentas de bioinformática estão publicamente disponíveis para predição e anotação de genes inseticidas, como NCBI Blast ou BtToxin_scanner, que usam sequências genômicas de proteínas de cristal de cepas de Bt. Uma vez que as ferramentas de bioinformática identifiquem o tamanho completo da(s) sequência(s) de codificação da toxina, é possível amplificar por PCR e clonar para expressar a proteína no sistema hospedeiro de escolha (PALMA et al., 2014; ADANG; CRICKMORE, 2014; YE et al., 2012), (subseção 2.4.2) .

2.5.1 O *Pipeline* Computacional BtToxin_scanner

O sistema *BtToxin_scanner* é uma ferramenta *web*[1] de mineração que tem dois objetivos (YE et al., 2012):

- obter a sequência genômica com plasmídeo enriquecido misto de Bt usando a biotecnologia do sequenciamento de nova geração;
- identificar genes *cry*.

Esse *pipeline* integra três tipos diferentes de métodos - BLAST (seção 3.4), HMM (seção 3.1) e SVM - para predizer a presença do gene da toxina Cry (YE et al., 2012). A construção do pipeline computacional utiliza módulos do BioPerl e está disponível para o uso de terceiros. Também está disponível um pequeno conjunto de dados curados para ser utilizado no método do *pipeline*. Os algoritmos utilizados na classificação e na predição de proteínas são:

- BLAST: utiliza métodos de alinhamentos locais emparelhados para medir a similaridade das sequências;
- HMM (*Hidden Markov Model*): utiliza métodos baseados em alinhamentos múltiplos gerados por um perfil estatístico HMM;
- SVM (*Support Vector Machine*): método que transforma sequências de proteínas em vetores de características de tamanho fixo.

Preprocessamento das Sequências de Entrada

As entradas podem ser sequências de nucleotídeos, proteínas ou ORFs (YE et al., 2012). Uma fase de leitura aberta ou ORF (do inglês *open reading frame*) consiste exclusivamente de triplas representações de aminoácidos. Uma sequência que é traduzida em proteínas tem uma ORF que inicia com um codon inicial especial (AUG) e se extende por uma série de triplas representando até o codon terminador (UAA) (LEWIN; LEWIN, 2008).

As entradas são convertidas em sequências de proteínas ao serem processadas. Para cada tipo de entrada existe um módulo de tradução para encontrar todas as sequências proteicas possíveis. Nesse processo, as sequências com tamanho inferior a 115 aminoácidos são eliminadas por não fornecerem informações para clonagem de genes *cry* (YE et al., 2012).

[1] Disponível no *site* http://bcam.hzaubmb.org/BtToxin_scanner/

Módulo de Predição do Candidato a *cry*

Os módulos BLAST, HMM e SVM predizem as sequências de genes *cry*. A ferramenta utiliza aprendizado de máquina com função de base radial (RBF, do inglês *radial basis function*) para o treinamento dos vários modelos (YE et al., 2012).

As proteínas com similaridade significativa menor que 1 e -30 são eliminadas para reduzir o número de falso-positivos das sequências de proteína Cry que surgiriam como resultado ao usar os três métodos de predição (YE et al., 2012).

O *BtToxin_scanner* utiliza dois conjuntos de dados, uma base de dados *cry* e uma base de dados de conhecimento. A base de dados *cry* é um pequeno conjunto de dados curados manualmente, que foi utilizado com o intuito de reduzir o nível de erros que existem em bases de dados públicas. Essa base de dados reduz o tempo necessário para o processo de predição em uma base de dados com todas as sequências de proteínas Cry. A base de dados *cry* e a base de dados de conhecimento são atualizadas regularmente (YE et al., 2012).

Segundo os resultados publicados (YE et al., 2012), foram extraídas sequências de 21 estirpes de Bt. Dessas sequências, 8 foram identificadas como genes *cry*, 3 identificadas como potenciais novos tipos de genes *cry*, e 5 delas se tornaram holótipos *cry*. Esse sistema para mineração de genes de Bt combina sequenciamento de genoma enriquecido com plasmídeos misturados e um *pipeline* computacional, minera novas sequências relacionadas à seleção de estirpe anterior. Também enfatiza que a seleção da estratégia de sequenciamento afeta os resultados da predição, variando de acordo com a escolha entre custo ou eficiência.

Apesar dos resultados acima apontarem a eficiência da ferramenta, é recorrente a instabilidade no acesso ao *site*. Por diversas vezes o ambiente está inacessível, impossibilitando qualquer condição de utilização dos recursos oferecidos.

3 FERRAMENTAS DE BIOINFORMÁTICA

Nas duas últimas décadas as tecnologias de autorendimento, como *microarray*, sequenciação e reação em cadeia de polimerase (PCR), foram amplamente aplicadas em pesquisas biológicas. Porém, com o aumento da quantidade de dados genômicos ficam os desafios de como gerenciá-los e quais métodos computacionais e ferramentas de análise disponíveis utilizar (D'ANTONIO et al., 2013; BAO et al., 2014).

O objetivo desse capítulo não é abordar essa discussão, mas apresentar quais os métodos selecionados para atender os objetivos desse trabalho.

Esse trabalho combina serviços de ferramentas de análise, onde o resultado de uma busca de similaridade de sequência, por exemplo, pode ser utilizada como a entrada para um alinhamento múltiplo de sequência, bastando para isso apenas informar o identificador do *job* entre os serviços utilizados (MCWILLIAM et al., 2013; LI et al., 2015).

Desde 1998, o Instituto Europeu de Bioinformática (EMBL-EBI, *European Bioinformatics Institute*) fornece acesso livre e aberto a uma variedade de aplicações de bioinformática para análise de sequências. A partir de 2004 o EMBL-EBI passou a fornecer acesso a uma variedade de bases de dados, que chegam a armazenar 20 petabytes de dados, além de ferramentas de análise por intermédio de interfaces Web Services [2]. Tais interfaces Web Services são baseadas nas tecnologias REST ou SOAP para utilizar as bases de dados e ferramentas que permitem a integração a outras ferramentas, aplicações, *web sites*, processos de *pipeline* e fluxos de trabalho analíticos, evitando, assim, a necessidade de manter essas bases de dados e programas localmente (MCWILLIAM et al., 2013; BAO et al., 2014; LI et al., 2015).

O termo *web service* pode significar qualquer serviço disponível na *World Wide Web*, ou ainda qualquer serviço que esteja baseado em tecnologias *web*, no qual se pretende utilizar programas de computador ao invés de pessoas, por fim, um *web service* pode ser utilizado para especificar tecnologias de serviços web, como SOAP e REST (WEB..., 2017).

Os Web Services apresentam uma abordagem modular e orientada a configuração, permitindo aos usuários executarem uma busca na base de dados e recuperarem os dados solicitados, retornando tais dados no formato específico, ou ainda, combinar serviços e criar *pipeline* de dados e fluxo de trabalhos analíticos (MCWILLIAM et al., 2013; LI et al., 2015).

As interfaces web e Web Services fornecidos são adaptados para as especificidades

[2] Disponível em http://www.ebi.ac.uk/Tools/webservices/

dos dados dos usuários, assim como a utilização de modelos necessários, permitindo que as entradas de dados sejam suportadas para uma variedade de tipos de formatos, como por exemplo o formato de sequência FASTA, XML, RDF/XML e SeqXML. As entradas de dados pelo *browser* são verificadas e todos os parâmetros necessários são validados, para que a submissão do *job* seja bem sucedida ou, em caso de falha, seja fornecida uma orientação ao usuário (MCWILLIAM et al., 2013; LI et al., 2015).

Os serviços de busca e recuperação de dados disponibilizam uma variedade de ferramentas de análise, incluindo serviços de busca de similaridade de sequência, alinhamento múltiplo de sequências, alinhamento pareado de sequências, análise funcional de proteínas, entre outros (MCWILLIAM et al., 2013; LI et al., 2015).

Os resultados podem ser recuperados em uma variedade de formatos gráficos e legíveis por máquina. O EMBL-EBI disponibiliza um guia de estilos[3]. As tabelas com os resultados sumarizados por busca de similaridade de sequência, por exemplo, podem ser disponibilizadas nos formatos XML, CSV, TSV e JSON. Para o caso de análise filogenética, a saída é uma matriz de identidade em porcentagem e a visualização de uma árvore através do uso de tecnologias JavaScript (LI et al., 2015).

É possível utilizar esses modelos em linha de comando, inclusive combinando os serviços de tal forma que os processos sejam executados em um único comando. No entanto, para consultas onde os requisitos de recuperação de dados sejam mais complexos, talvez seja necessário que o usuário do serviço use os modelos dentro de um *script* com comandos adicionais utilizando linguagens de programação comuns, além da possibilidade em obter os resultados enriquecidos com novas visualizações. Nesses casos a implementação e o uso desses Web Services demanda uma lógica adicional (MCWILLIAM et al., 2013; LI et al., 2015).

O EMBL-EBI disponibiliza um breve guia[4] dos Web Services, possibilitando aos desenvolvedores aprenderem mais, integrarem funcionalidades adicionais aos programas e web sites que desenvolvem, sem a necessidade de se preocuparem em manter cópia das bases de dados ou softwares envolvidos (MCWILLIAM et al., 2013; LI et al., 2015).

3.1 HMMER: Procura Iterativa de Similaridade de Sequência

A busca por similaridade em base de dados de sequência é uma das aplicações mais importantes envolvendo a biologia molecular computacional. As sequências genômicas estão sendo adquiridas rapidamente para um conjunto cada vez maior de espécies. As ferramentas computacionais de comparação por homologia permitem utilizar o máximo de dados de sequências, aprender pistas sobre suas funções desconhecidas e histórias evolutivas

[3] Acessível em https://ebiwd.github.io/EBI-Pattern-library/
[4] Disponível em http://www.ebi.ac.uk/Tools/webservices/tutorials/00_contents.

(WISTRAND; SONNHAMMER, 2005; EDDY, 2011).

Nesse contexto, HMMER é um pacote de software *open-source* desenvolvido por Sean Eddy para a detecção de sequências homólogas de proteínas utilizando métodos probabilísticos avançados por intermédio de várias ferramentas úteis (WISTRAND; SONNHAMMER, 2005; FINN; CLEMENTS; EDDY, 2011).

O HMMER é utilizado para construir modelos de busca em base de dados de alinhamentos preexistentes, como as bases de dados de família de proteínas Pfam, TIGRFRAMs e SMART. No entanto, é possível criar modelos como *motif*, que constituem abordagens muito poderosas para a descoberta de *motifs* conservados em conjunto de sequências inicialmente não alinhadas (EDDY, 1998; WISTRAND; SONNHAMMER, 2005).

O alinhamento múltiplo de sequências pode fornecer posições específicas que tendem a ser mais conservadas quando comparadas a outras. Por exemplo, o alinhamento múltiplo de sequências de diferentes toxinas Cry revelaram a presença de até cinco blocos conservados localizados no núcleo ativo tóxico das proteínas (PALMA et al., 2014; SCHNEPF et al., 1998; BRAVO et al., 1998). O perfil HMM é um modelo computacional que faz uso dessa informação para a representação do alinhamento de sequências referentes a uma mesma família (TANIGUTI, 2014).

O HMMER está disponível nas versões *Web Service*[5] e de linha de comando UNIX, uma versão local. Os módulos que compõem o HMMER constroem modelos e alinhamento de sequências de DNA ou proteína. É possível construir um perfil HMM (modelo oculto de Markov) de uma entrada de alinhamento múltiplo, ou alinhamento múltiplo de muitas sequências para um perfil HMM comum. Os alinhamentos múltiplos de sequências são importantes em aplicações para estimar a árvore filogenética, a predição estrutural e identificação crítica de resíduos (EDDY, 1992; EDGAR, 2004b; WISTRAND; SONNHAMMER, 2005; FINN; CLEMENTS; EDDY, 2011).

Ao se trabalhar com sequências de DNA, como é a proposta desse trabalho, o HMMER disponibiliza alguns programas. Especificamente a implementação da ferramenta utiliza o programa *nhmmscan*, no entanto, esse programa faz uso de uma base de dados de perfil HMM e, para gerar essa base no formato necessário, é preciso executar outros programas que fazem parte do pacote HMMER conforme apresentado na Tabela 1.

3.1.1 Formatos Suportados pelo HMMER

Geralmente, o HMMER[6] pode detectar automaticamente o formato de um alinhamento de sequências múltiplas e o formato de um arquivo de sequência não alinhado (EDDY, 1992). Abaixo alguns dos vários formatos que o HMMER lê:

[5] Disponível em http://www.ebi.ac.uk/Tools/hmmer/
[6] Disponível em http://hmmer.org/

Tabela 1 – Programas disponibilizados pelo HMMER que trabalham com sequências de DNA (EDDY, 1992)

Programa	Função
hmmbuild	Constroi um perfil HMM para uma entrada de alinhamento múltiplo.
nhmmscan	Pesquisa uma sequência de DNA contra uma base de dados de perfil HMM.
hmmpress	Formata uma base de dados HMM para um formato binário para ser utilizando pelo *nhmmscan*.

- *sto* no formato estocolmo (do inglês *Stockhom format*). Nesse formato os pareamentos de bases são identificados pelos símbolos ( ) (parênteses), [ ] (colchetes) e { } (chaves). As bases únicas e resíduos são representados pelos símbolos _ (sublinhado), - (hífen), , (vírgula), : (dois pontos), . (ponto) e ˜ (til) (OLIVEIRA, 2009);

- *hmm* no formato texto ASCII;

- *fa* no formato FASTA;

- *out* gera um arquivo de saída.

O arquivo de saída é uma tabela com 16 colunas de informações que estão descritos na Tabela 2 (EDDY, 1998).

Tabela 2 – Colunas da tabela de saída de pesquisa por DNA utilizando o HMMER

Campo	Descrição
Target name	O nome do perfil
Accession	A adesão do perfil caso exista, ou "-" se não existe
Query name	O nome da do perfil de consulta
Accession	A adesão do pefil de consulta caso exista, ou "-" se não existe
HMMfrom	A posição no *hmm* na qual o *hit* inicia
HMM to	A posição no *hmm* na qual o *hit* termina
Alifrom	A posição inicial do *hit* na sequência alvo
Ali to	A posição final do *hit* na sequência alvo
Envfrom	A posição na sequência alvo na qual inicia o envelope proteico
Env to	A posição na sequência alvo na qual termina o envelope proteico
Modlen	O tamanho da sequência alvo
Strand	A cadeia na qual o *hit* foi encontrado
E-value	A significância estatística da sequência alvo
Score	A pontuação de acerto para o *hit*
Bias	A correção de composição tendenciosa para geração do *score*
Description of target	Texto livre de descrição da sequência alvo

3.1.2 Perfil HMM

Desde 1994 as arquiteturas HMM são modelos probabilísticos que representam adequadamente perfis de alinhamentos de sequências múltiplas ou individuais. Os perfis utilizam o alinhamento dessas sequências para capturar informações sobre uma posição específica, como cada coluna conservada do alinhamento e quais são os prováveis resíduos, para representar a família nas pesquisas na base de dados (EDDY, 1992; WISTRAND; SONNHAMMER, 2005).

Modelos de perfil HMM (*profile HMM*) utilizam dois algoritmos de programação dinâmica, um para alinhamento (*Viterbi*) e outro para pontuação (*Forward*), com o intuito da detecção de homologia em sequências. O perfil HMM típico é uma cadeia de nós representando os estados de *match* (acertos), *insert* (inserções) e *delete* (exclusões), onde o único melhor caminho corresponde a um caminho do estado inicial até o estado final, em que cada caractere da sequência está relacionado a sucessivos estados de *match* e *insert* ao longo do caminho. Vale ressaltar, no entanto, que o algoritmo calcula um custo na transição entre estados sempre que não haja contrapartida, ou não possua um caractere correspondente a essa posição no HMM (EDDY, 1998; KARPLUS; BARRETT; HUGHEY, 1998; JOHNSON; EDDY; PORTUGALY, 2010).

Todos os métodos de perfil de alinhamentos de sequências múltiplas utilizam pontuação de posição específica de aminoácidos ou nucleotídeos (resíduos) e penalidades de posição específica para abertura e extensão de uma inserção ou exclusão. HMMs têm uma base probabilística formal, onde para cada coluna de consenso do alinhamento múltiplo, um estado de *match* modela a distribuição de resíduos permitidos na coluna. Um estado de *insert* permite a inserção de um ou mais resíduos entre aquela coluna e a próxima, ou *delete* para deleção do resíduo de consenso de cada coluna (EDDY, 1992; EDDY, 1998).

3.1.3 Pesquisando uma Base de Dados de Perfil HMM uma Sequências de DNA

O comando *hmmbuild* constrói um perfil HMM de um ou de múltiplos alinhamentos a partir de um arquivo que esteja em um formato válido. O resultado é um arquivo de perfil(s) HMM (EDDY, 1992).

Ao invés de buscar em um modelo único contra uma coleção de sequências, o usuário pode querer anotar todas as instâncias de uma coleção de perfis HMM encontrados em uma única sequência. O *nhmmscan* é utilizado para pesquisar sequências contra coleções de perfis. No caso de DNA o *nhmmscan* é funcional para anotação de todos os domínios conhecidos ou detectáveis em uma dada sequência. A base de dados utilizada poderia ser de perfis HMM para famílias de elementos transponíveis (Dfam), porém, nesse trabalho foi utilizada uma base de dados de elementos regulatórios conservados (EDDY, 1992).

Para criar uma base de dados é necessário construir arquivos HMM individuais e então concatená-los ou utilizar alinhamentos que estejam em formatos aceitos pelo HMMER e executar o programa *hmmbuild* para construir a base de dados. Uma outra opção, ainda, é concatenar todos os arquivos de alinhamentos juntos e construir os arquivos HMM para todos os alinhamentos de uma vez (EDDY, 1992).

Os arquivos HMM gerados pelo HMMER estão em código ASCII e são muito volumosos. O programa *nhmmscan* tem a função de ler rapidamente vários desses arquivos. O modo mais eficiente de realizar essa leitura é ler esses arquivos no formato binário ao invés de lê-los em código ASCII. Então, para empregar o *nhmmscan* é necessário primeiro comprimir e indexar a base de dados HMM com o programa *hmmpress* (EDDY, 1992). O resultado da execução do *hmmpress* gera quatro arquivos proprietários, descritos a seguir:

- Arquivo *.h3m* contém os perfis HMM e suas anotações em um formato binário.
- Arquivo *.h3i* é um índice SSI para o arquivo *.h3m*.
- Arquivo *.h3f* contém as estruturas de dados précomputadas para o filtro heurístico rápido (filtro MSV).
- Arquivo *.h3p* contém as estruturas de dados précomputadas para o resto de cada perfil.

Essa etapa é realizada com o intuito de obter a base de dados HMM. A repetição dessa etapa voltaria a acontecer somente em uma atualização no conjunto de arquivos de alinhamentos que compõem a base de dados HMM.

A partir desse ponto é possível analisar sequências utilizando a base de dados HMM e o programa *nhmmscan*. A saída produzida leva em consideração o *E-value*, além dos *scores*, domínio, coordenadas de domínio e alinhamento. O HMMER usa algoritmos de amostragem estocástica para inferir alguns parâmetros, e mesmo assim, a cada execução sempre mostrará o mesmo resultado para o mesmo problema.

3.2 BEDTools: Manipulação de Intervalo Genômico

BEDTools é um poderoso conjunto de ferramentas utilizadas para uma ampla variedade de tarefas de análise genômicas ou aritmética de genoma. A primeira versão foi disponibilizada em 2009, desenvolvida pelo laboratório Quinlan na Universidade de Utah. A motivação pelo seu desenvolvimento deve-se a necessidade de ferramentas que fossem rápidas e flexíveis o suficiente para comparar grandes conjuntos de características genômicas. A continuidade do conjunto das ferramentas é beneficiada por contribuições da comunidade científica mundial (QUINLAN; KINDLON, 2017).

BEDTools é uma ferramenta de linha de comando que permite realizar a interseção (*intersect*), combinação (*merge*), contagem (*count*), complemento (*complement*), e o embaralhamento (*shuffle*) de intervalos genômicos de múltiplos arquivos que estejam em uma ampla variedade de formatos de arquivos genômicos. Cada ferramenta individualmente realiza uma tarefa relativamente simples, porém, se combinadas as múltiplas operações podem realizar análises bastante sofisticadas. (QUINLAN; KINDLON, 2017; QUINLAN, 2017).

As operações suportadas pela BEDTools permitem a manipulação de características genômicas. As características genômicas podem ser elementos funcionais, polimorfismo genético e anotações que têm sido descobertas ou curadas por grupos de sequenciamento genômico, ou anotações personalizadas quem um laboratório individual ou pesquisador define (QUINLAN; KINDLON, 2017).

Entre as características básicas de um genoma estão os cromossomos, no qual residem as características, os pares de base inicial e terminal da característica, a vertente (*strand*) no qual a característica existe (+ ou –) e o nome da característica se aplicável (QUINLAN; KINDLON, 2017).

O conjunto de ferramentas BEDTools permite comparar características contidas em dois arquivos distintos. Duas características genômicas se sobrepõem ou se cruzam se elas compartilham pelo menos uma base em comum (QUINLAN; KINDLON, 2017; QUINLAN, 2017).

Na ferramenta desenvolvida nesse trabalho foram necessárias as ferramentas:

- *intersect*: para encontrar os intervalos de sobreposição.
- *merge*: para combinar os intervalos sobrepostos em um um único intervalo.

3.2.1 Comparar Características

O comando *intersect* é o principal do conjunto BEDTools, e compara dois ou mais arquivos a fim de identificar todas as regiões no genoma onde as características nos dois arquivos se sobrepõem, ou seja, regiões que compartilham pelo menos um par de base em comum (QUINLAN, 2017).

Por padrão, o *intersect* mostra os intervalos que representam sobreposição entre dois arquivos A e B, como ilustra a Figura 5. Mas, a partir da versão 2.21.0, o BEDTools permite realizar o *intersect* de um arquivo contra um ou vários arquivos. A incorporação desse recurso permite uma análise simplificada envolvendo múltiplas bases de dados relevantes para um mesmo experimento (QUINLAN; KINDLON, 2017; QUINLAN, 2017). Nesse trabalho utilizamos a versão 2.25.0 do BEDTools.

Figura 5 – Comportamento padrão do comando *intersect*. Fonte: (QUINLAN; KINDLON, 2017).

Além da execução padrão é possível utilizar outras opções que geram relatórios com as características originais em cada arquivo, ou a quantidade de pares de base que se sobrepuseram, ou a contagem do número de características de sobreposição e a comparação de vários arquivos de uma única vez (QUINLAN, 2017).

3.2.2 Combinar Intervalos

Muitos *datasets* de características genômicas têm muitas características individuais para sobrepor uma a outra. O comando *merge* do BEDTools permite combinar apenas a sobreposição dentro de um intervalo contíguo único (QUINLAN, 2017).

Para a execução do comando *merge* é necessário que o arquivo de entrada esteja ordenado por cromossomo e depois pela posição inicial. Se os arquivos não estiverem ordenados o comando *merge* retornará um erro. A ordenação permite que o algoritmo de mesclagem trabalhe muito rapidamente sem requerer a utilização da memória. Para ordenar o arquivo podemos utilizar o comando *sort* no terminal UNIX (QUINLAN, 2017).

O comando *merge* combina os resultados em um novo conjunto de intervalo, representando o conjunto combinado de intervalos na entrada. Ou seja, como ilustrado na Figura 6, se um par de base no genoma é coberto por 10 características, ele agora será representado uma única vez no arquivo de saída (QUINLAN, 2017).

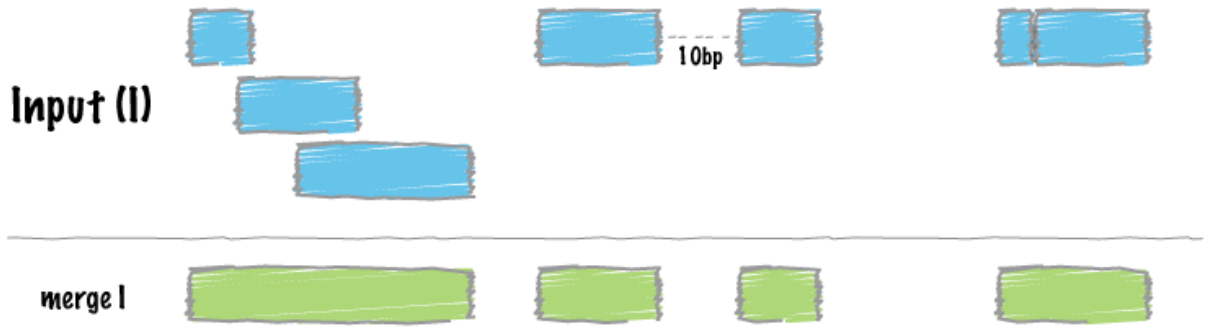


Figura 6 – Comportamento padrão do comando *merge*. Fonte: (QUINLAN; KINDLON, 2017).

Além do fato de sobrepor intervalos combinados, é possível utilizar outras opções mais sofisticadas, como por exemplo, gerar um relatório com o número de intervalos que foram integrados ao novo intervalo combinado, ou combinar intervalos que não se sobrepõem, ainda que estes estejam próximos um do outro e criar uma lista dos nomes de cada característica (QUINLAN, 2017).

3.3 MUSCLE: Alinhamento de Sequências Múltiplas

O crescimento das bases de dados de sequências de famílias de proteínas excede a capacidade da maioria das ferramentas de análise. Obter a precisão biológica dos alinhamentos e atender a complexidade computacional no que se refere a tempo de execução e requisitos de memória são atributos primordiais para os programas de alinhamento de múltiplas sequências (EDGAR, 2004a).

A formulação mais natural do problema computacional é definir um modelo de evolução sequencial que atribui probabilidades para todas as possíveis edições de sequências elementares e buscar a sequência mais provável em um grafo dirigido, no qual as arestas representam edições e os nós terminais são as sequências observadas. Esse grafo pode ser interpretado como uma árvore filogenética e implica um alinhamento, não sendo considerados separadamente (EDGAR, 2004a; EDGAR, 2004b).

Nenhum método acessível é reconhecido por encontrar um grafo ideal para modelos biológicamente realistas e, portanto, é necessária fazer a simplificação. O MUSCLE é uma ferramenta para o alinhamento de sequências múltiplas de nucleotídeos ou aminoácidos. O MUSCLE[7] é um programa de computador para criar alinhamentos múltiplos e reconstrução de árvore filogenética, que fornece melhorias significativas tanto na acurácia quanto na velocidade. O Instituto Europeu de Bioinformática oferece acesso ao MUSCLE como uma API suportada pelo protocolo REST, por exemplo (EDGAR, 2004a; EDGAR, 2004b; MUSCLE. . . , 2017).

O algoritmo inclui estimativa de distância rápida usando contagem k-mer, alinhamento progressivo empregrando uma função de perfil chamada *log-expectation* e refinamento, utilizando particionamento restrito dependente de árvore (EDGAR, 2004b).

3.3.1 Implementação

A estratégia utilizada pelo MUSCLE consiste na construção de um alinhamento progressivo, que aplica um refinamento horizontal. O algoritmo tem três estágios, como ilustrado na Figura 7. Ao completar cada estágio, um alinhamento múltiplo é disponibilizado e o algoritmo pode ser finalizado (EDGAR, 2004a).

[7] Disponível em http://www.ebi.ac.uk/Tools/msa/muscle/

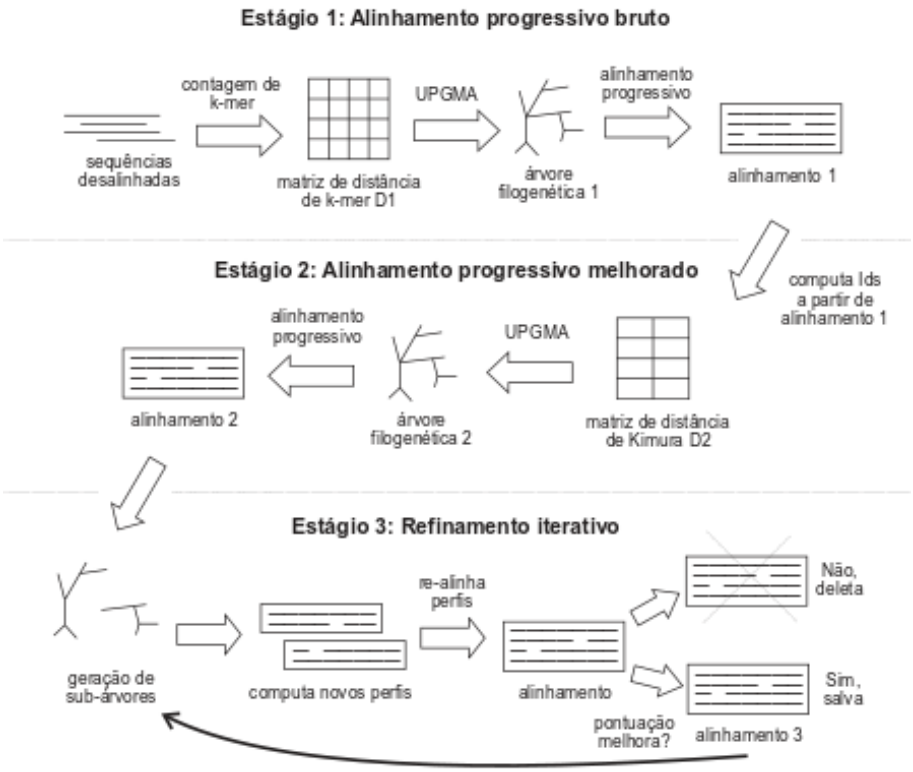


Figura 7 – Diagrama de fluxo do algoritmo de três estágios utilizado pelo MUSCLE. Fonte: (MARUCCI, 2009).

O algoritmo MUSCLE utiliza duas medidas de distância para um par de sequências. São elas a distância k-mer para um par desalinhado e a distância Kimura para um par alinhado. A distância k-mer é uma sequência contígua de tamanho k, conhecido como uma palavra (*word*) ou k-tupla. Essa medida não requer um alinhamento. A distância Kimura é aplicada para corrigir múltiplas substituições em um único local, quando é dado um par de sequências alinhadas e deseja se calcular a identidade emparelhada (EDGAR, 2004b; SIMPSON et al., 2009).

As matrizes de distâncias geradas são clusterizadas utilizando um dos dois métodos disponíveis: o método UPGMA ou o método *Neighbor-Joining* (EDGAR, 2004a; EDGAR, 2004b).

Alinhamento Progressivo Bruto

O primeiro estágio constrói um alinhamento progressivo com base na medida de similaridade, estimativa de distância, construção da árvore e o alinhamento progressivo (EDGAR, 2004a). Nesse primeiro estágio o objetivo é produzir um alinhamento múltiplo enfatizando mais a velocidade do que a precisão (EDGAR, 2004b).

A similaridade de cada par de sequência de entrada é calculada considerando a contagem de k-mer (EDGAR, 2004a; EDGAR, 2004b), gerando a matriz de distância D1

apresentada na Figura 7.

Uma matriz de distância triangular D1 é estimada a partir das similaridades dos pareamentos. Essa matriz de distância é clusterizada pelo método UPGMA ou *Neighbor-Joining*, produzindo uma árvore binária chamada na Figura 7 de árvore filogenética 1 (EDGAR, 2004a; EDGAR, 2004b), e uma raiz é identificada.

Um alinhamento progressivo é construído seguindo as ordens de ramificações da árvore, produzindo um alinhamento múltiplo de todas as sequência de entrada na raiz (EDGAR, 2004a). A cada folha, um perfil é construído a partir de uma sequência de entrada. Os nós na árvore são visitados em ordem de prefixo, ou seja, os filhos são visitados antes dos pais. Produz um alinhamento múltiplo de todas as sequências de entrada na raiz, onde, para cada nó interno é construído um alinhamento pareado de dois perfis filhos. Na Figura 7, no estágio 1, é chamado de alinhamento 1 (EDGAR, 2004b).

Alinhamento Progressivo Melhorado

A principal fonte de erro no primeiro estágio é a medida de distância aproximada k-mer, que resulta em uma árvore subótima. O segundo estágio tenta melhorar a árvore e construir um novo alinhamento progressivo de acordo com essa árvore, utilizando a medida de distância Kimura, que é mais precisa por requerer um alinhamento. O segundo estágio é iterativo e conta com a medida de similaridade, a construção e comparação da árvore e o alinhamento progressivo (EDGAR, 2004a; EDGAR, 2004b).

A similaridade de cada par de sequências é obtida usando cálculo de identidade fracional de alinhamentos mútuos no alinhamento múltiplo atual. Esse método consiste em receber como entrada apenas o alinhamento resultante do primeiro estágio, no caso, o alinhamento 1. A sua execução consiste em dois laços aninhados que calculam a distância entre todos os pares da sequência (EDGAR, 2004a; MARUCCI, 2009), resultando na matriz de distância D2, apresentada na Figura 7.

A árvore é construída pelo cálculo da matriz de distância D2 e é aplicado um dos métodos de clusterização a essa matriz (EDGAR, 2004a), produzindo uma árvore binária, na Figura 7 chamada de árvore filogenética 2.

A árvore criada no primeiro estágio é comparada com essa nova árvore, identificando o conjunto de nós internos para o qual a ordem da ramificação mudou. O número de alterações na árvore dependerá de quantas iterações acontecerá no segundo estágio (EDGAR, 2004a). Ou seja, um alinhamento progressivo é produzido seguindo a árvore filogenética 2, produzindo um alinhamento múltiplo chamado na Figura 7 de alinhamento 2.

Os alinhamentos existentes são mantidos e novos alinhamentos são criados para o conjunto de nós que sofreram mudanças. Quando o alinhamento da raiz estiver completo,

o algoritmo pode terminar, retornando ao início do segundo estágio ou indo para o terceiro estágio (EDGAR, 2004a).

Refinamento Iterativo

O terceiro estágio executa um refinamento iterativo considerando o particionamento. O processo consiste na escolha de uma bipartição, da extração do perfil e seu realinhamento, além de uma pontuação capaz de manter ou descartar o novo alinhamento (EDGAR, 2004a).

Uma bipartição consiste na exclusão de uma aresta da árvore, dividindo as sequências em dois subconjuntos disjuntos. Ou seja, uma aresta da árvore filogenética 2 é escolhida e as arestas são visitadas em ordem da menor distância a partir da raiz. A árvore filogenética 2 é dividida em duas subárvores por exclusão de arestas, e o perfil de cada subconjunto é extraído do alinhamento múltiplo atual, sendo que as colunas que não contém resíduos são descartadas (EDGAR, 2004a; EDGAR, 2004b).

Os dois perfis obtidos nessa etapa são realinhados, produzindo um novo alinhamento múltiplo. Se a pontuação aumentou, o novo alinhamento é mantido, caso contrário é descartado, como mostra a Figura 7. O algoritmo finaliza se todas as arestas foram visitadas e não houve troca, caso contrário retorna para o início do terceiro estágio (EDGAR, 2004a; EDGAR, 2004b).

A finalização do algoritmo pode acontecer por convergência ou até chegar em um limite definido pelo usuário. Esse estágio é uma variante do particionamento restrito dependente de árvore.

3.3.2 Métodos para Construção de Árvores Filogenéticas

Os relacionamentos evolutivos entre organismos são sumarizados em diagramas chamados árvores filogenéticas, ou simplesmente, filogenias. Essas árvores podem tanto mostrar os relacionamentos entre os organismos como sobrepor os relacionamentos em uma linha de tempo para indicar como cada organismo evoluiu (SNUSTAD, 2015).

Entender uma filogenia é bem parecido com ler uma árvore genealógica. A raiz da árvore representa a linhagem ancestral, e as pontas das ramificações representam os descencentes desse ancestral. Quando uma linhagem se divide chama-se especiação, e é representada como uma ramificação na filogenia. Quando o evento de especiação ocorre, uma única linhagem ancestral dá origem a duas ou mais linhagens filhas (ENTENDENDO..., 2016).

Na construção de árvores filogenéticas são utilizados três métodos estatísticos: parcimônia e verossimilhança, baseados em caracteres na árvore, e baseados em distância, como *Neighbor-Joining* e UPGMA (EDGAR, 2004a). Os métodos baseados em distância

comparam sequência homólogas de DNA reduzindo a variação entre estas a uma única medida de distância, resultando na árvore filogenética (LOPES, 2006).

Muitos métodos estão disponíveis para construir árvores filogenéticas a partir de dados de sequências de DNA ou proteínas. Abaixo apresenta- se quatro características comuns a esses métodos (SNUSTAD, 2015):

1. Alinhar as sequências para permitir comparações entre elas;
2. Determinar a quantidade de similaridade (ou diferença) entre quaisquer duas sequências;
3. Agrupar as sequências com base na similaridade;
4. Colocar as sequências nas pontas de uma árvore.

Os métodos de distância requerem duas etapas: o cálculo da distância, que gera uma matriz de distância, e a construção da topologia. A classificação das proteínas Cry utiliza o método do vizinho mais próximo (MAAGD; BRAVO; CRICKMORE, 2001).

Método UPGMA

O algoritmo UPGMA (do inglês *Unweighted Pair Group Method with Arithmetic means*) é o método mais simples de construção de topologias de cladogramas moleculares a partir de dados de distância. Esse método tenta estimar uma árvore de espécies ou a árvore gênica esperada (BUSO, 2005; MACIEL, 2011; LOPES, 2006). Esse método leva em consideração o coeficiente de correlação das frequências gênicas entre colônias. O modelo assume que uma população inteira é subdivida em colônias e que a migração de indivíduos em cada geração é restrita às colônias próximas, chamadas de dimensões. A diminuição da correlação genética com distância depende muito do número de dimensões (KIMURA; WEISS, 1964).

UPGMA emprega um algoritmo sequencial de agrupamento de pares não ponderados, baseado na média aritmética. Nele, as relações são identificadas em ordem de similaridade, e leva em consideração a menor distância (BUSO, 2005; MACIEL, 2011; LOPES, 2006). A árvore é construída passo a passo:

1. Indentifica-se entre todas as OTUs[8] (*Operational Taxonomic Unit*) estudadas as duas mais similares e aí trata como se fosse uma unidade.
2. Do resto do grupo identifica-se outra unidade com maior similaridade.

[8] Uma OTU é um agrupamento de *reads* com 97% de similaridade, motivado por uma expectativa que esse agrupamento corresponda a espécies.

3. Volta para o passo 1 e continua até o final.

Esse método pode ser usado na construção de árvores filogenéticas sempre que as distâncias utilizadas refletirem uma certa proporcionalidade com o tempo de evolução. Quando a distância estimada está sujeita a grandes erros estocásticos, o UPGMA é superior a outros métodos que também utilizam matrizes de distância para recuperar a árvore.

Método Neighbor-Joining

O termo *neighbors* (vizinhos) é um par de OTUs conectadas por intermédio de um único nó interior não raiz, resultando em uma ramificação. O número de pares de vizinhos em uma árvore depende da topologia da árvore. Por exemplo, se N representa a quantidade de OTUs, e N for par, o número máximo de vizinhos será N/2, porém, caso N seja ímpar, o número máximo de vizinhos será (N - 1)/2 (SAITOU; NEI, 1987).

O método *Neighbor-Joining* foi proposto para reconstrução de árvores filogenéticas a partir de qualquer tipo de dado de distância evolucionária. O princípio desse método é encontrar pares de OTUs que minimizem o comprimento total do ramo a cada estágio do agrupamento dos vizinhos, começando com uma árvore pequena em forma de estrela (SAITOU; NEI, 1987), como na Figura 8.

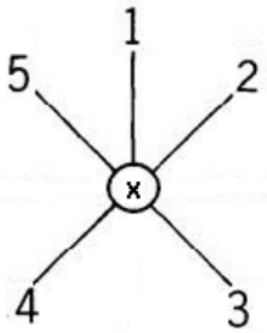


Figura 8 – Exemplo da representação do estado da árvore inicial utilizando o algoritmo de *Neighbor-Joining*. Fonte: (CARRIÇO,).

Em seguida, procura-se o par de vizinhos que minimize a soma total dos ramos da árvore, gerando a primeira bifurcação e serão tratados como uma única unidade, como na Figura 9, onde os táxons 2 e 3 foram agrupados. E assim o processo continua até que se obtenha uma árvore completamente resolvida (LOPES, 2006; CARRIÇO,).

Os métodos tradicionais são demorados, e, quando o número de OTUs é grande, somente uma pequena proporção de todas as topologias possíveis é examinada. O método *Neighbor-Joining*, em compensação, produz uma árvore final única sob o princípio da evolução mínima. Esse método não necessariamente produz a árvore de evolução mínima, mas é eficiente em obter a topologia correta da árvore (SAITOU; NEI, 1987).

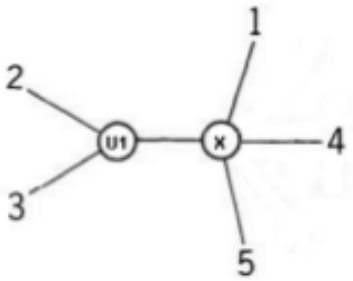


Figura 9 – Exemplo da representação da primeira bifurcação após clusterizar dois vizinhos pelo algoritmo de *Neighbor-Joining*. Fonte: (CARRIÇO,).

3.4 BLAST: Ferramenta de Busca de Alinhamento Básico Local

O BLAST é uma ferramenta de comparação de sequências e pesquisa em base de dados, utilizando uma abordagem heurística para construir alinhamentos, otimizando uma medida de similaridade local, baseada em pontuações de mutações. Por intermédio de um algoritmo de programação dinâmica detecta similaridades de sequências com significados biológicos (ALTSCHUL et al., 1990; TATUSOVA; MADDEN, 1999; EDDY, 2011).

Um alinhamento consiste em realizar a comparação de duas ou mais sequências fazendo com que a posição de cada base nitrogenada ou aminoácido dessas sequências fique uma sob a outra (LOPES, 2006). No entanto, pela presença de *indels*, inserções e ou deleções de nucleotídeos ou mutações, pode vir a gerar sequências homólogdas de DNA de tamanhos diferentes. Por essa razão ocorre a inserção de *gaps* em determinados pontos, de modo que as sequências apresentem bases igualmente alinhadas. O método de alinhamento procura valorizar as bases igualmente alinhadas e penalizar alinhamentos de bases desiguais ou com *gaps* (LOPES, 2006; EDGAR, 2004a).

Um alinhamento pode ser pensado de tal forma a minimizar a distância evolucionária ou maximizar a similaridade entre duas sequências comparadas. Nesse caso, o custo desse alinhamento é uma medida de similaridade; o algoritmo garante o ideal baseado em pontuações (*scores*) (ALTSCHUL et al., 1990).

O BLAST encontra múltiplos alinhamentos locais entre duas sequências, permitindo ao usuário detectar domínios proteicos ou duplicações de sequências internas. Também é útil na comparação de genes homólogos a partir de genomas microbianos completos. O uso do BLAST para comparação de sequência de nucleotídeos de cepas diferentes ou isolados do mesmo vírus é uma estratégia para estudar as variações de genomas e eventos evoluicionários, assim como substituições, inserções e deleções (TATUSOVA; MADDEN, 1999).

Uma versão do programa na *web* pode ser utilizada a partir do site do NCBI[9]. Os resultados dos alinhamentos são apresentados tanto na forma gráfica quanto textual. As

[9] Disponível em https://blast.ncbi.nlm.nih.gov/Blast.cgi

variações do programa para PC estão disponíveis nas plataformas Windows, Mac e Linux, e podem ser baixados, também, a partir do site do NCBI[10] (TATUSOVA; MADDEN, 1999).

3.4.1 Métodos

As medidas de similaridade de sequências geralmente podem ser classificadas como global ou local. Algoritmos de similaridade global otimizam o alinhamento total das duas sequências, que pode incluir grandes extensões e baixa similaridade. Já os algoritmos de similaridade local buscam somente subsequências relativamente conservadas, e uma simples comparação pode render vários alinhamentos de subsequências distintas (ALTSCHUL et al., 1990).

Muitas medidas de similaridade começam com uma matriz de similaridade de pontuações para todos os possíveis pares de resíduos. Um segmento de sequência é uma extensão contígua de resíduos de qualquer tamanho, e a pontuação de similaridade para dois segmentos alinhados do mesmo tamanho é a soma dos valores de similaridade para cada par de resíduos alinhados (ALTSCHUL et al., 1990).

MSP: A Medida Pares de Segmento Máximo

Um par de segmento máximo é a maior pontuação de um par de segmentos de tamanhos idênticos escolhidos de duas sequências. A pontuação MSP para duas sequências pode ser computada utilizando um algoritmo de programação dinâmica. A partir da matriz de pontuação pode estimar as frequências de resíduos emparelhados nos segmentos máximos (ALTSCHUL et al., 1990).

Em uma pesquisa em uma base de dados com milhares de sequências, o cientista está interessado em identificar sequências que compartilham similaridade altamente significativa com a sequência de consulta. O BLAST minimiza o tempo gasto na pesquisa a base de dados, procurando apenas pares de segmentos que contenham um par de palavra com uma pontuação maior ou igual ao limite defnido que é o tamanho da palavra (número de letras) (ALTSCHUL et al., 1990; TATUSOVA; MADDEN, 1999).

Implementação do Algoritmo

A implementação dessa abordagem detalha 3 passos que variam um pouco dependendo se a base de dados contém proteínas ou sequências de DNA (ALTSCHUL et al., 1990).

- compilar uma lista de palavras de alta pontuação;

[10] Disponível em https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE_TYPE=BlastDocs&DOC_TYPE=Download

- varredura da base de dados na busca por acertos;
- estender acertos.

O algoritmo e estatísticas implementadas pelo BLAST permite emparelhar localmente sequências de DNA-DNA ou proteína-proteína (ALTSCHUL et al., 1990; TATUSOVA; MADDEN, 1999).

Para proteínas, a lista consiste de todas as palavras que pontuam pelo menos um limite T quando comparadas a alguma palavra na sequência da consulta. Assim, uma palavra de consulta pode ser representada por nenhuma palavra na lista ou por muitas (ALTSCHUL et al., 1990).

Para DNA, consiste em uma lista de palavras mais simples, lista todas as palavras contíguas na sequência de consulta. As sequências de DNA são altamente não-aleatórias, com composição de base tendenciosa localmente, e elementos de sequência repetidos, com importante consequência para o projeto de uma ferramente de pesquisa em base de dados de DNA (ALTSCHUL et al., 1990).

O número de letras padrão para alinhamentos de nucleotídeos-nucleotídeos é 11, e para alinhamentos proteína-proteína o valor padrão é 3, além de, em ambos alinhamentos, ser necessário uma combinação exata do tamanho da palavra entre as duas sequências. Para se alcançar uma melhor sensitividade é possível utilizar uma quantidade menor de letras para o tamanho das palavras, respeitando a restrição de intervalo de 7 a 20 para nucleotídeos e de 2 a 3 para proteínas (TATUSOVA; MADDEN, 1999).

3.4.2 Algoritmos do BLAST

O BLAST encontra regiões de similaridade local entre sequências. O programa compara sequências de nucleotídeos ou proteínas em base de dados de sequências e calcula o significado estatístico dos acertos. O BLAST pode ser utilizado para inferir relacionamentos funcional e evolucionário entre sequências, e ajudar na identificação de membros de famílias de genes (MADDEN, 2013).

Há diferentes pesquisas usando BLAST para todas as necessidades (MADDEN, 2013):

- BLAST nucleotídeos: pesquisa uma base de dados de nucleotídeos usando uma consulta de nucleotídeo. Utiliza os algoritmos: BLASTn e megaBLAST.
- BLAST proteínas: pesquisa uma base de dados de proteínas usando uma consulta de proteína. Utiliza os algoritmos: BLASTp, psi-BLAST e delta-BLAST.
- BLASTx: pesquisa em uma base dados de proteínas usando uma consulta de nucleotídeos traduzidos.

- tBLASTn: pesquisa uma base de dados de nucleotídeos traduzidos usando uma consulta de proteína.
- tBLASTx: pesquisa uma base de dados de necleotídeos traduzidos usando uma consulta de nucleotídeos traduzidos.

4 *PIPELINE PARA DESCOBERTA E CATEGORIZAÇÃO DE GENES CRY*

Esse capítulo tem por objetivo descrever o *pipeline* e as suas funcionalidades, que são alcançadas por meio do uso combinado de programas computacionais de bioinformática. Esse trabalho apresenta o desenvolvimento de um *web site* utilizando software *open source* de bioinformática, como HMMER, BEDTools, BLAST e MUSCLE descritos no Capítulo 3, com a finalidade de identificar e classificar genes *cry*, recebendo como entrada uma sequência de nucleotídeos (seção 1.1).

O *pipeline* é o resultado da reunião de esforços em uma parceria com o Programa de Pós-Graduação em Bioinformática da Universidade Tecnológica Federal do Paraná e o laboratório de Bioinformática da Universidade Estadual de Londrina, e conta com a supervisão dos professores Dr. Alessandro Botelho Bovo e Dr. Laurival Antônio Vilas Boas, além da colaboração da Msc. Katia Brumatti Gonçalves, que construiu a base de dados curada de genes *cry*, e o Msc. Ivan Rodrigo Wolf, responsável por desenvolver um *Shell script* que combina intervalos de características genômicas com a finalidade de estreitar os resultados da busca de sequências de nucleotídeos em uma base de dados de perfil HMM.

A relevância desse trabalho está diretamente relacionada às necessidades de se encontrar novas toxinas Cry com toxicidade melhorada para o controle de pragas que são controladas de forma ineficiente (seção 1.2).

4.1 O *Pipeline* Computacional Proposto

O *pipeline* consiste na combinação de ferramentas de bioinformática, com o intuito de encontrar de maneira putativa novas toxinas Cry, apresentando o alinhamento das sequências com a melhor correspondência e a reconstrução de árvores filogenéticas para encontrar o(s) parente(s) mais próximo(s), a partir de uma entrada contendo uma ou várias sequências de nucleotídeos.

Para o *pipeline*, basicamente o usuário acessa o navegador e digita o endereço do *site* para acessá-lo. As interfaces do *website* são projetadas em HTML e CSS, já a parte dinâmica do *site* utiliza a linguagem Python. O Python através do *framework* Flask e do módulo *subprocess* acessa a base de dados dos principais software de bioinformática utilizados nesse trabalho, e retorna para o navegador os resultados da busca.

Tabela 3 – Software relacionados ao trabalho

Serviço	Descrição de uso	Utilização
HMMER	Detecção de sequências homólogas de nucleotídeos	Local
BEDTolls	Análise genômica	Local
NCBI BLAST	Comparação e alinhamento de sequências pesquisando em base de dados de nucleotídeos	Web Service
MUSCLE	Alinhamento de sequências múltiplas e reconstrução da árvore filogenética	Web Service

O *framework* Flask[11] é uma micro estrutura de desenvolvimento *web* para o Python[12]. Esse *framework* visa manter funcionalidades simples e ao mesmo tempo extensíveis, ou seja, fáceis de serem modificadas. O Flask é micro no sentido em que não oferece funcionalidades para banco de dados e validação de formulários, por exemplo, mas oferece suporte para que tais funcionalidades sejam adicionadas à uma aplicação como se fossem implementadas pelo próprio Flask (RONACHER, 2017).

Os Web Services do BLAST e do MUSCLE foram acessados por meio do pacote Python BioServices[13], que fornece acesso a base de dados ou aplicações de bioinformática via uma interface web utilizando tecnologias baseada nos protocolos SOAP e REST. A tecnologia REST não tem dependência externa, já que utiliza URLs, retornando um documento no formato XML ou TSV para ser analisado (COKELAER et al., 2016).

A Tabela 3 apresenta os software que serão utilizados nesse trabalho. Enquanto que a Figura 10 ilustra uma abstração que retrata as etapas de execução do *pipeline*. Na figura também é possível observar a atividade fundamental de gerar o perfil *hmm* descrito na subseção 4.3.1 e que está fora do escopo do *pipeline*.

[11] http://flask.pocoo.org/
[12] https://www.python.org/
[13] https://pythonhosted.org/bioservices/

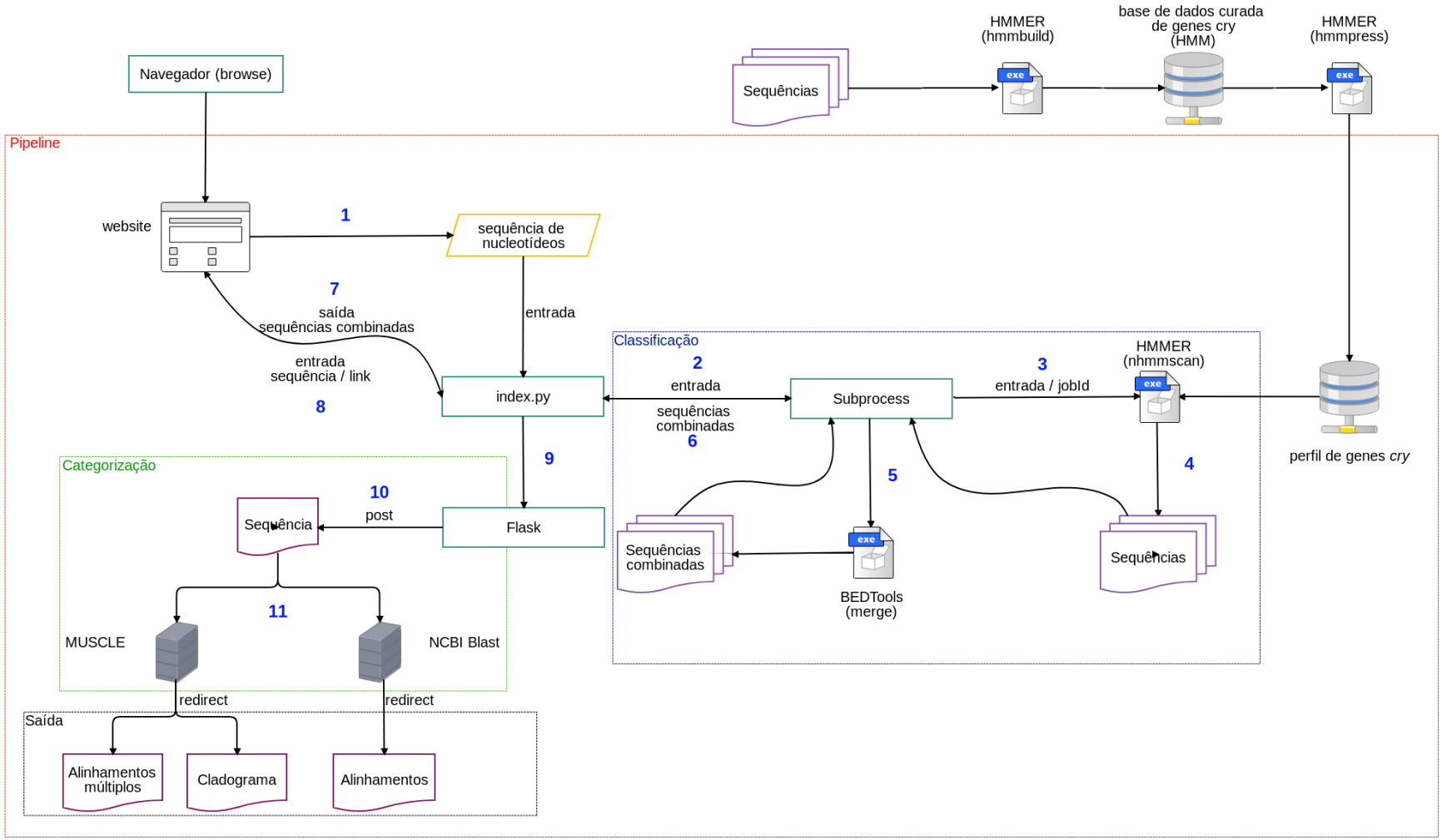


Figura 10 – Fluxograma do *pipeline* para categorizar e classificar possíveis genes *cry*.

Para uma melhor compreensão da sequência do fluxo do *pipeline*, as etapas no diagrama ilustradas na Figura 10 foram numeradas na ordem de execução:

1. Ao acessar o *website* o usuário deve informar a sequência de nucleotídeos. Essa entrada corresponde a um arquivo no formato FASTA.

2. A aplicação, identificada por *index.py*, recebe esse arquivo que serve de parâmetro na chamada do módulo *subprocess*. Esse módulo permite gerar um novo processo local, recebendo dados pela entrada padrão e retornando um resultado.

3. O módulo *subprocess* passa como parâmetros de entrada para o processo *nhmmscan* o arquivo FASTA, a identificação do *job* (que resultará no arquivo de saída) e a base de dados com os perfis de genes *cry*.

4. O *nhmmscan*, como explicado na subseção 3.1.3, retorna uma tabela com todas as sequências possíveis considerando o arquivo de entrada contra a base de dados.

5. Essas sequências servem de entrada para o processo *merge* que está encapsulado em um *Shell script*. Como explicado na subseção 3.2.2, o *merge* irá combinar todas as sequências que tenham características genômicas comuns em um único intervalo, restringindo o resultado do *nhmmscan*.

6. O módulo *subprocess* retorna as sequências combinadas para a aplicação *index.py*.

7. A função da aplicação nesse momento é processar os dados resultantes da etapa 6 e exibí-las ao usuário no *website*. O usuário saberá a localização gênica, incluindo informações estatísticas e *links* para acessar informações resultantes do BLAST e ou do MUSCLE.

8. As informações serão paginadas em um *grid* com 10 resultados. O usuário pode navegar entre as páginas e selecionar o *link* para o BLAST ou MUSCLE (um de cada vez) referente a sequência selecionada.

9. A aplicação receberá como entrada a localização gênica e em qual programa realizará a pesquisa.

10. O *framework* Flask faz a conexão com os *webservices* do BLAST ou do MUSCLE e suas respectivas bases de dados. Os parâmetros são passados utilizando o método *post*, onde são informadas a sequência alvo, a indicação do programa e a base de dados ou formato de saída, dependendo do programa.

11. As saídas serão redirecionadas para o próprio *website* das aplicações BLAST ou MUSCLE, e o usuário poderá ver e analisar os alinhamentos, a matriz de distâncias, o cladograma entre outras informações disponíveis.

A sequência desse capítulo apresenta cada etapa que compõe esse *pipeline* no intuito de encontrar e classificar um candidato a gene *cry*.

4.2 Processo de Cura da Base de Dados de Genes *cry*

O objetivo dessa seção é descrever o processo de cura da base de dados utilizada no *pipeline*. A lista completa do genes de Bt, que codificam as δ-endotoxinas, pode ser obtida a partir do "*Bacillus thuringiensis* toxin nomenclature" (CRICKMORE et al., 2016), responsável pela nomeclatura das toxinas Cry. A partir dessa coleta foram feitos os alinhamentos levando-se em consideração até o terceiro ranque da nomenclatura.

A nomenclatura de uma proteína Cry possui quatro níveis de classificação (subseção 2.2.1). Tomando como exemplo a proteína Cry1Aa, o mnemônico (Cry) indica o tipo de proteína, o número (1) é a classificação primária, a letra maiúscula (A) é a classificação secundária, e a letra minúscula (a) é a classificação ternária. Há ainda uma quarta classificação, representada por um número, que indica o compartilhamento mais de 95% de identidade na sequência de aminoácidos da família.

A partir desses alinhamentos foram feitas as anotações curadas. O BLAST (seção 3.4) foi usado para confirmar se as sequências estavam corretas. Às vezes acontece de uma determinada sequência de um mesmo grupo ser diferente das demais, ou por início diferente (ATG) ou pelo comprimento. Com o BLAST foi possível confirmar essas informações. Utilizando o UGENE foi possível encontrar as ORFs, localizando-se o códon de iniciação e o códon de terminação do gene (seção 2.5.1), retirando-se o que não interessava.

A gestão, análise e visualização dos dados biológicos ficou a cargo do software Unipro UGENE. Esse software inclui várias características descritas (ALGAER, 2016; OKONECHNIKOV et al., 2012), como:

- Criar, editar e anotar ácidos nucléicos e sequências de proteínas.
- Pesquisar em bases de dados online: NCBI, SWISS-PROT, etc.
- Alinhamento de sequência múltipla: ClustalW, ClustalO, Muscle, etc.
- Pesquisa local e online BLAST.
- Pesquisa por ORFs.
- Integração com os pacotes HMMER2 e HMMER3.
- Construir e visualizar árvores genéticas.

Com o UGENE foi possível gerar os arquivos (FASTA) com as sequências de nucleotídeos. A base de dados curada conta com um total de 699 sequências de genes *cry*.

O processo de cura da base de dados envolveu a obtenção dos dados públicos e o trabalho de anotação desses dados, resultando em uma outra base de dados de genes *cry*. Essa etapa gera a *base de dados curada de genes cry* ilustrada na Figura 10.

Outros trabalhos nessa linha de pesquisa, como o BtToxin_scanner, trabalham com sequências de aminoácidos, já nesse trabalho foi utilizada uma base de dados de nucleotídeos com as sequências de genes *cry* conhecidas atualmente. Isso torna necessário que essa base de dados curada seja atualizada periodicamente. Sendo que esta atividade ficará na responsabilidade dos pesquisadores no laboratório de Bioinformática da Universidade Estadual de Londrina.

4.3 Etapas do Pipeline

Nas próximas seções serão descritas cada uma das etapas do pipeline. Tais estapas buscam desenvolver um fluxo de trabalho para análise e detecção de genes *cry*. Sempre que um novo gene *cry* for identificado e categorizado, recebendo sua devida nomenclatura e tendo ou não sua toxicidade confirmada, tal sequência deve ser incluída na base de dados e passar pelo processo de cura, conforme descrito anteriormente.

4.3.1 Geração do Perfil HMM

A geração do perfil HMM acontece sem a participação direta do usuário. Esse processo fundamental no desenvolvimento do trabalho é ilustrado na Figura 10, e está fora do escopo do *pipeline*. A atualização do perfil será realizada periodicamente, porém esse processo não está automatizado.

Para cada arquivo de alinhamento múltiplo de sequência o programa *hmmbuild* constrói um perfil HMM, salvando-o em um novo arquivo, num dos formatos suportados pelo HMMER indicados em subseção 3.1.1 (EDDY, 1992). O perfil HMM será utilizado para discriminar por intermédio do alinhamento múltiplo de sequências membros da família de Bt revelando as regiões conservadas que são características de determinada família.

O exemplo do comando a seguir gera um perfil HMM:

```
$ hmmbuild Cry1Ae.hmm Cry1Ae.fasta
```

A construção do modelo HMM foi realizada a partir da base de dados curada, que usa DNA como tipo de alfabeto. Uma base de dados HMM é simplesmente uma concatenação de arquivos HMM individuais.

Exemplo do comando para concatenar todos os arquivos HMM, gerando um único arquivo:

```
$ cat CryDB >> *
```

As sequências que compõem a base de dados curada são representadas por códigos ASCII, então, o programa *hmmpress* concatena esses arquivos e representa-os em códigos binários, gerando o perfil HMM necessário para ser utilizado pelo *nhmmscan*, conforme descrito na subseção 3.1.3.

Exemplo do comando que *hmmpress* que prepara a base de dados HMM para ser utilizada pelo *nhmmscan*:

```
$ hmmpress CryDB
```

4.3.2 Busca de Sequências de Nucleotídeos em um Perfil de Nucleotídeo

A partir dessa etapa o processo está automatizado no *pipeline*. A primeira etapa corresponde a entrada ilustrada na Figura 10, e consiste na obtenção das sequências que serão comparadas com o perfil HMM. Sendo assim, a ferramenta recebe como entrada uma sequência FASTA e, a partir dessa entrada, visa classificar sequências que caracterizem famílias de genes *cry*. FASTA é o formato padrão baseado em texto para representar dados de sequências biológicas. Um arquivo com o formato FASTA começa com o símbolo > e o nome da sequência, seguido por linhas com letras, que no caso desse trabalho representam nucleotídeos (BAO et al., 2014).

A Figura 11 apresenta a tela inicial, onde o usuário deverá informar o arquivo contendo a(s) sequência(s).

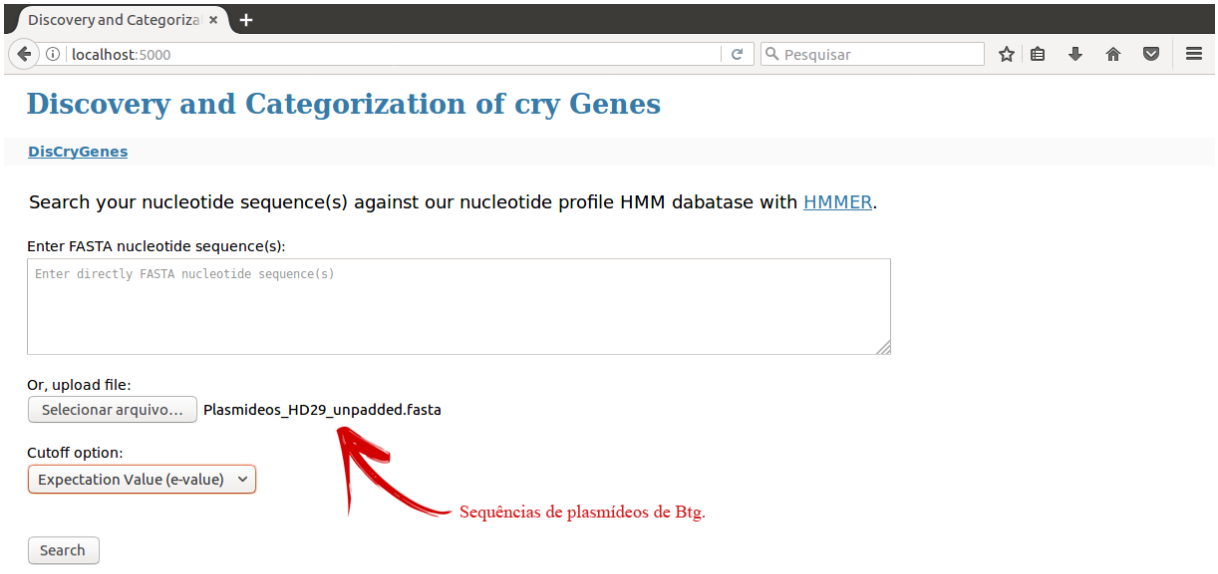


Figura 11 – Tela inicial da ferramenta de categorização.

Ao clicar no botão *Search* é iniciado o programa *nhmmscan* para pesquisar cada sequência de entrada na base de dados de perfis e fornecer como saída uma lista classificada

dos perfis com as combinações mais significativas para a sequência (EDDY, 1992), como é possível verificar na Figura 10.

A função *start_hmmer* utiliza o módulo *subprocess*, indicado na etapa 2 da Figura 10, que permite executar um processo, conectando seus fluxos de entrada, saída e erro, além de obter o retorno do código. A criação e o gerenciamento do processo *nhmmscan* são tratados pela classe *Popen*. Alguns argumentos da classe foram utilizados no excerto de código a seguir, como PIPE para capturar o erro padrão (*stderr*) e *communicate* que retorna uma tupla (dados da saída padrão e dados de erro padrão).

O processo *nhmmscan* passa para o argumento *–tbout* o arquivo de saída, que é representado por *USER_FILE_PATH + job_id*, além da base de dados utilizada, que é indicada por *HMMER_PROFILE*, e qual é o arquivo que foi selecionado pelo usuário, identificado por *USER_FILE_PATH + job_id + '.fasta'*, como indicado na etapa 3 do *pipeline* ilustrado na Figura 10.

```
def start_hmmer(job_id):
    try:
        exit_code = 0 #No error
        p = subprocess.Popen(['nhmmscan',
                              '--tblout',
                              USER_FILE_PATH + job_id,
                              HMMER_PROFILE,
                              USER_FILE_PATH + job_id + '.fasta'],
                              stderr=subprocess.PIPE)
        stdout, stderr = p.communicate()
        exit_code = p.poll()
        if stderr!=None:
            print(stderr.decode('utf-8'))
    except Exception as e:
        print(e)
        exit_code = 1
        #To do: log the error info
    finally:
        Config.set_job_end(USER_FILE_PATH, job_id, exit_code)
    print('fim start_hmmer')
```

Ao executar o *nhmmscan* é gerado um *job id* e informado um endereço ao usuário onde será possível acompanhar o andamento do processo e verificar o resultado do *job*, exatamente como ilustra a Figura 12. Após o término da execução, para efeito de controle da ferramenta são gerados três arquivos: a arquivo FASTA com as sequência alvo, a tabela com o resultado do *nhmmscan* e um arquivo com o tipo *.config*, que apresenta três colunas,

as duas primeiras tem o início e fim, respectivamente, e a última coluna tem um valor que pode ser 0 (terminou sem erros) ou 1 (terminou com erros).

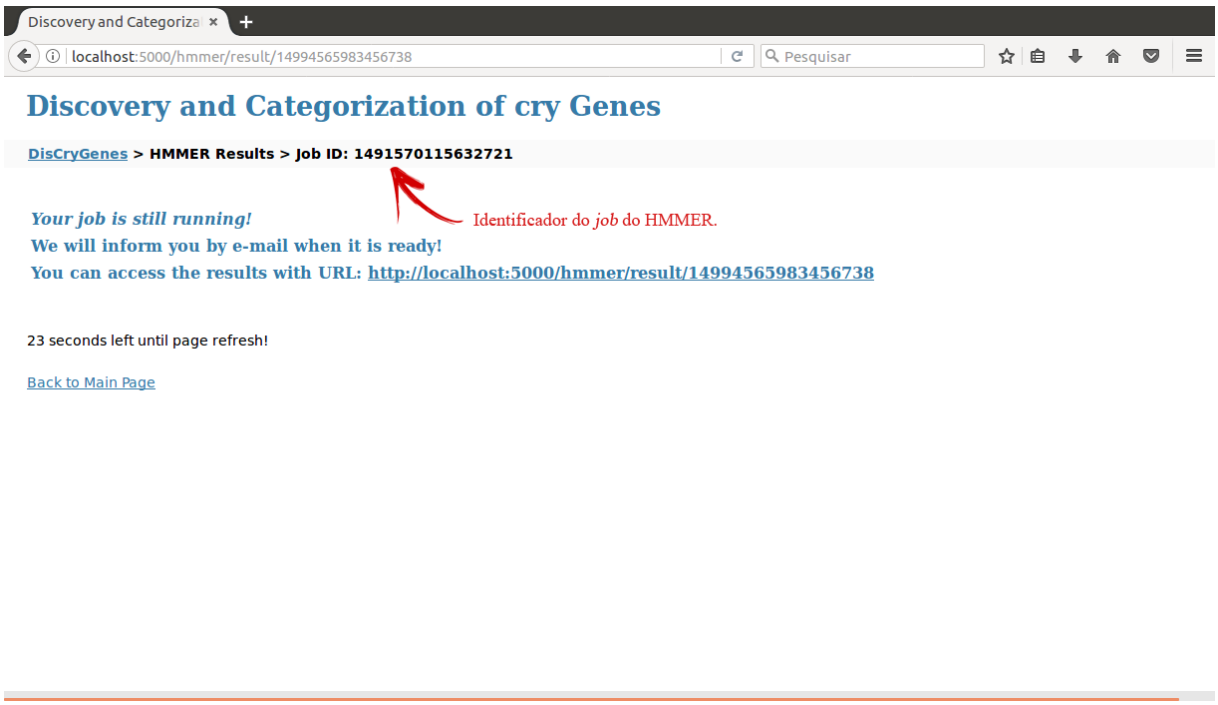


Figura 12 – Andamento do *job* desencadeado pelo *nhmmscan*.

A quarta etapa do *pipeline* é a saída produzida pelo *nhmmscan* está em um formato tabular, legível para os usuários, mas normalmente gera uma volume muito grande de informação, tornando a leitura impraticável e dificultando a análise. Por essa razão, a ferramenta utiliza um *script* capaz de filtrar os resultados combinando intervalos que se repetam de forma contígua nas várias linhas de saída, restringindo o resultado.

4.3.3 Obter o Conjunto Combinado de Intervalos

A quinta etapa da execução extrai informações da tabela de resultados produzida pelo HMMER, identificando o *locus* e a classificação do gene *cry* putativo.

O *script* utiliza o processo *merge* do BEDTools que, como visto anteriormente na subseção 3.2.2, necessita que as colunas da tabela de saída do HMMER sejam reordenadas. Então, no arquivo de entrada do *script* as colunas *alifrom* e *ali to*, as colunas 7 e 8 respectivamente, trocam de posição, como exibido no excerto de código a seguir:

```
cat $in_file | grep -v "#" | grep '+' | awk '{OFS="\t"}
 {print $1,$2,$3,$4,$5,$6,$7,$8,$9,$10,$11,$12,$13,$14,$15,$16}'
 | awk -v min_len=$min_len '{OFS="\t"}
 {if (($8 - $7) > min_len) print $0;}' > $out_file.tbl.tmp
wait
cat $in_file | grep -v "#" | grep -v '+' | awk '{OFS="\t"}
```

```
{print $1,$2,$3,$4,$5,$6,$8,$7,$9,$10,$11,$12,$13,$14,$15,$16}'
| awk -v min_len=$min_len '{OFS="\t"}
{if (($8 - $7) > min_len) print $0;}' >> $out_file.tbl.tmp
wait
```

O comando *cat* exibe o conteúdo de um arquivo, asssim como, permite que um arquivo receba o conteúdo de outro, nesse caso utiliza os caracteres especiais '>>'. O comando *grep* por sua vez, procura em um ou mais arquivos por linhas que contenham um padrão de busca ou, no caso do uso da opção *-v*, exibe linhas que não contenham o padrão '#'. Na saída do HMMER as duas primeiras linhas começam com o símbolo '#' e, portanto, serão desconsideradas. O comando ou linguagem *awk* é uma ferramenta de análise para criar filtros de conteúdos de arquivos (JARGAS, 2008; NEVES, 2008). Também é possível ver que o *script* pode trabalhar com um tamanho mínimo de gene, porém, como os tamanhos dos genes *cry* variam na base de dados, limitaria ainda mais a lista de resultados. Por essa razão, na execução padrão, esse parâmetro não obrigatório é ignorado.

Na sequência do *script* cria-se uma lista de *contigs* com o correspondente gene *cry*. Lê cada *contig* e cria um arquivo *bed*, descrito na subseção 3.2.2, recuperando e salvando cada localização dos *contigs*.

```
while read contig
do
  cat $out_file.tbl.tmp | awk '{OFS="\t"}{print $3,$7,$8;}' |
     grep "$contig" | sort -k2 -n |
     bedtools merge -i stdin >> $out_file.gene.locus.lst
  wait
done <<< "$contigs"
wait
```

O arquivo de localização é composto de 3 colunas, *query name*, *alifrom* e *ali to*. O comando *sort -k2 -n* ordena o arquivo pela segunda coluna, indicada pela opção *-k2*, e como trata-se de um campo que contém números, a opção *-n* organiza esses números em ordem aritmética (JARGAS, 2008; NEVES, 2008).

O último arquivo gerado pelo *script*, e que é apresentado pela ferramenta ao usuário, mostra os *contigs* e a sua localização, além de informações adicionais, como *target name*, *score* e *bias*, descritas na Tabela 2 da subseção 3.1.1.

```
while read id start end
 do
 cat $out_file.tbl.tmp | grep "$id" | awk -v s=$start -v
   e=$end -v id=$id '{OFS="\t"}
```

```
    {if (($7 >= s) && ($8 <= e))
    print id,"CRY_DETECT","gene",s,e,$13,$12,".",
    "Name=" $1,"Bit-Score:" $14,"Bias:" $15;}' > $id.tmp
  wait
  cat $id.tmp | sort -k12 -r -n | head -n 1 >> $out_file.detected
  wait
  rm $id.tmp
  wait
done < $out_file.gene.locus.lst
wait
```

A Figura 13 representa a sétima etapa do *pipeline*, que consiste em exibir os melhores *hits* para o usuário. Nessa figura, a coluna Target Name corresponde ao nome do perfil apontado pelo *script* como candidato a gene *cry*. Cada *hit* apresentado na saída representa localização da sequência alvo na entrada. Por exemplo, na Figura 13 é possível verificar que a sequência de nucleotídeos consultada, localizada entre as posições 10454 e 11260, referente ao plasmídeo identificado por NODE_110_length_12556_cov_314.781 combina características genômicas com a família de gene Cry30Aa. Da mesma forma subsequência seguinte, localizada entre as posições 287 e 2187, poderia representar um gene da família Cry2Aa, e assim sucessivamente para cada *hit*. As colunas *ali from* e *ali to* fornecem as coordenadas do alinhamento local com relação ao perfil da consulta e a sequência alvo, respectivamente. (EDDY, 1992).

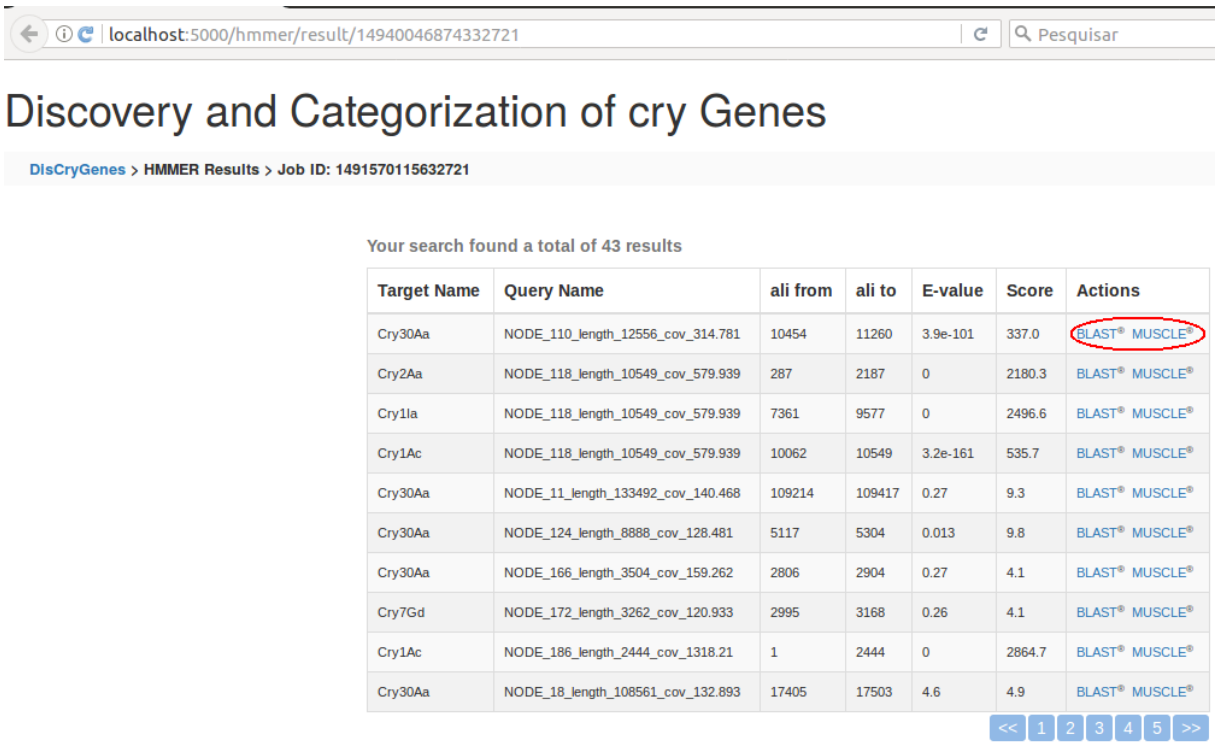


Figura 13 – Tela de saída na ferramenta de categorização.

A lista dos melhores *hits* ou acertos é colocada em ordem crescente por *E-value* (EDDY, 1992). O *E-value* (*expect value*) é um parâmetro baseado na sequência de *bit score* que descreve o número de *hits* que se pode esperar ver por acaso quando pesquisado em uma base de dados de tamanho específico. O *E-value* diminui exponencialmente quando o *score* por acerto aumenta. Quanto mais baixo for o *E-value*, tendendo a zero, mais significativo é o acerto.

A Figura 13 não apresenta a colula correspondente ao *E-value*, mas a coluna *score* em ordem decrescente. O *score* considera o *E-value*. No exemplo utilizado, o menor *E-value* começa com zero, assim, para todos os *E-value* valendo zero, a coluna *score* está classificada em ordem descrescente.

A partir da saída ilustrada na Figura 13, o usuário pode selecionar os *links*, indicados dentro da elipse vermelha, que inicializa a oitava etapa do *pipeline*, que será descrita nas seções subsequentes, e resultará na reconstrução da árvore filogenética e a apresentação dos alinhamentos das sequências.

4.3.4 Classificação Cladística

Os alinhamentos múltiplos de sequências são cruciais para o processamento de dados biológicos, possibilitando estimar a árvore filogenética, além da indicação crítica de resíduos. O excerto de código a seguir mostra alguns parâmetros necessários na décima etapa, como o formato de saída, o conjunto de sequências, além do uso de um e-mail padrão para que o usuário seja notificado quando os resultados estiverem disponíveis. Esses serão informados ao *web service* no momento em que o usuário selecionar o *link* MUSCLE.

```
@app.route('/muscle/<int:count>')
def muscle(count):
    try:
        if 'job_id' in session:
          job_id = session['job_id']
          fasta_file = USER_FILE_PATH + job_id + '.fasta'
          tab_out = USER_FILE_PATH + job_id
          hmm_result_list = Hmmer.process_result
                              (fasta_file, tab_out, 10)
          m = MUSCLE(verbose=False)
          #Convert to FASTA format in order to send to MUSCLE
          sequences = '>\n' + hmm_result_list[count].sequence +
              '\n' + hmm_result_list[count].curated_sequences
          muscle_job_id = m.run(frmt='clw',
                          sequence=sequences,
                          email='erinaldosnascimento@gmail.com')
```

```
            url = MUSCLE_WS + muscle_job_id
            print(url)
        else:
            url = url_for('index')  #To do: error messsage
    except:
        url = url_for('index')#To do: error messsage
    return redirect(url)
```

A sequência escolhida pelo usuário que está destacada com a borda em vermelho na Figura 14, representa o alinhamento da sequência alvo, e que pode indicar um gene que codifica a proteína Cry. O MUSCLE, então, apresenta os alinhamentos múltiplos e a árvore filogenética relacionados a essa sequência. A Figura 14 ilustra um cladograma construído por intermédio do método *Neighbor-joining*, de acordo com as opções de parâmetros passadas pela ferramenta para o *webservices*.

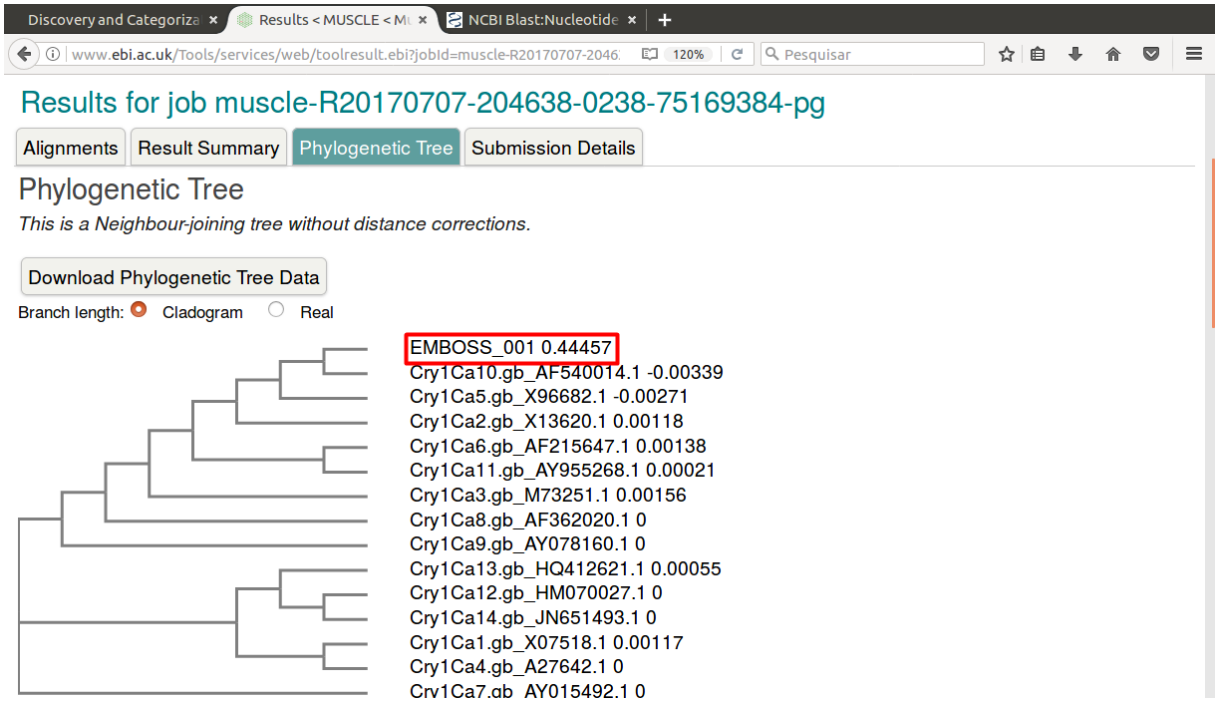


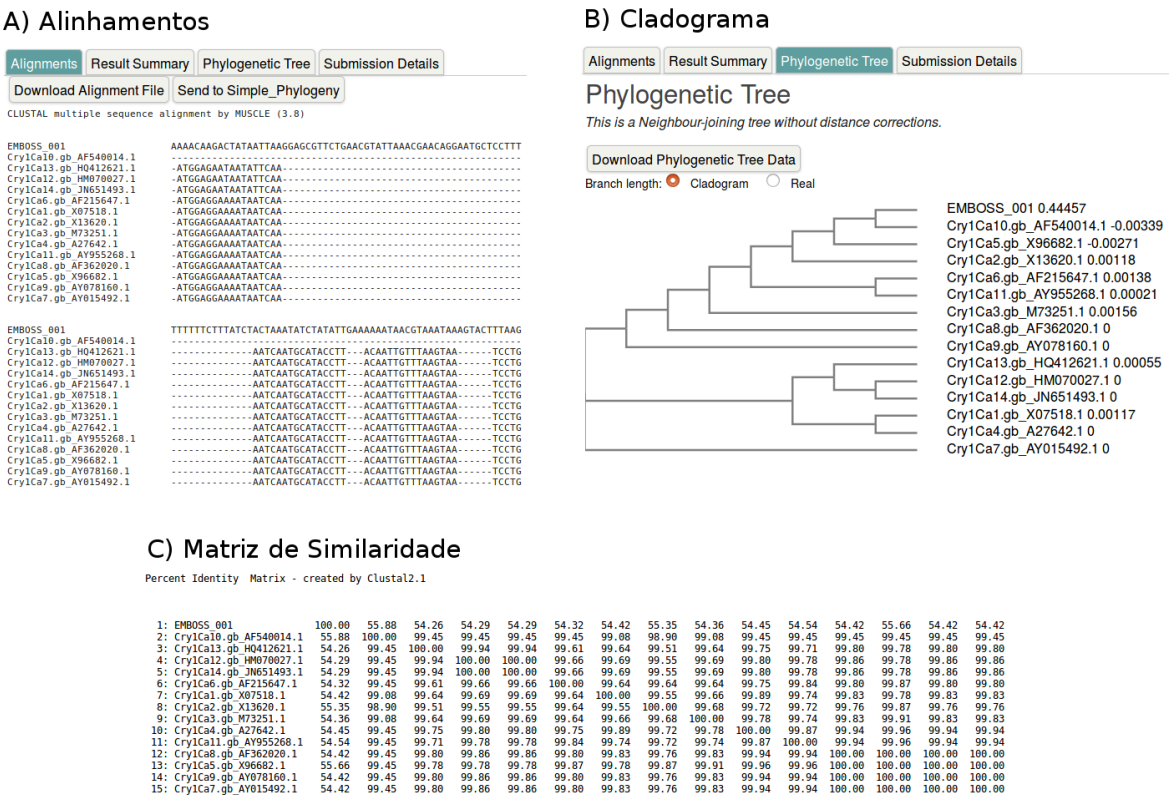
Figura 14 – Árvore filogenética resultante do método *Neighbor-Joining*.

Ao submeter um *job* é necessário informar tais parâmetros para o BioServices do MUSCLE. Os parâmetros obrigatórios são o formato, *frmt* que foi configurado com "*clw*"(ClustalW), e a sequência de consulta (*sequence*) no formato *fasta* recomendado (COKELAER et al., 2016).

Quando um *job* é submetido para execução o usuário pode obter informações por intermédio da identificação do *job*.

O MUSCLE apresenta ao usuário os alinhamentos das sequências e a árvore filogenética, partindo de uma matriz de distância, como ilustra a Figura 15. A saída do

MUSCLE possibilita ao usuário estabelecer as relações de ancestralidades da sequência escolhida.



C) Matriz de Similaridade

Percent Identity Matrix - created by Clustal2.1

1: EMBOSS_001	100.00	55.88	54.26	54.29	54.29	54.32	54.42	55.35	54.36	54.45	54.54	54.42	55.66	54.42	54.42
2: Cry1Ca10.gb_AF540014.1	55.88	100.00	99.45	99.45	99.45	99.45	99.08	98.90	99.08	99.45	99.45	99.45	99.45	99.45	99.45
3: Cry1Ca13.gb_HQ412621.1	54.26	99.45	100.00	99.94	99.94	99.61	99.64	99.51	99.64	99.75	99.71	99.80	99.78	99.80	99.80
4: Cry1Ca12.gb_HM070027.1	54.29	99.45	99.94	100.00	100.00	99.66	99.69	99.55	99.69	99.80	99.78	99.86	99.78	99.86	99.86
5: Cry1Ca14.gb_JN651493.1	54.29	99.45	99.94	100.00	100.00	99.66	99.69	99.55	99.69	99.80	99.78	99.86	99.78	99.86	99.86
6: Cry1Ca6.gb_AF215647.1	54.32	99.45	99.61	99.66	99.66	100.00	99.64	99.64	99.64	99.75	99.84	99.80	99.87	99.80	99.80
7: Cry1Ca1.gb_X07518.1	54.42	99.08	99.64	99.69	99.69	99.64	100.00	99.55	99.66	99.89	99.74	99.83	99.78	99.83	99.83
8: Cry1Ca2.gb_X13620.1	55.35	98.90	99.51	99.55	99.55	99.64	99.55	100.00	99.68	99.72	99.72	99.76	99.87	99.76	99.76
9: Cry1Ca3.gb_M73251.1	54.36	99.08	99.64	99.69	99.69	99.64	99.66	99.68	100.00	99.78	99.74	99.83	99.91	99.83	99.83
10: Cry1Ca4.gb_A27642.1	54.45	99.45	99.75	99.80	99.80	99.75	99.89	99.72	99.78	100.00	99.87	99.94	99.96	99.94	99.94
11: Cry1Ca11.gb_AY955268.1	54.54	99.45	99.71	99.78	99.78	99.84	99.74	99.72	99.74	99.87	100.00	99.94	99.96	99.94	99.94
12: Cry1Ca8.gb_AF362020.1	54.42	99.45	99.80	99.86	99.86	99.80	99.83	99.76	99.83	99.94	99.94	100.00	100.00	100.00	100.00
13: Cry1Ca5.gb_X96682.1	55.66	99.45	99.78	99.78	99.78	99.87	99.78	99.87	99.91	99.96	99.96	100.00	100.00	100.00	100.00
14: Cry1Ca9.gb_AY078160.1	54.42	99.45	99.80	99.86	99.86	99.80	99.83	99.76	99.83	99.94	99.94	100.00	100.00	100.00	100.00
15: Cry1Ca7.gb_AY015492.1	54.42	99.45	99.80	99.86	99.86	99.80	99.83	99.76	99.83	99.94	99.94	100.00	100.00	100.00	100.00

Figura 15 – Resultados obtidos pelo MUSCLE.

Por padrão a árvore filogenética utiliza o método *Neighbor-Joining*, porém o usuário pode alterar os parâmetros e reconstruir a árvore filogenética utilizando o método UPGMA. Isso é possível porque ao terminar a execução do MUSCLE fica liberado um link para que o usuário visualize os resultados obtidos.

4.3.5 Obter o Alinhamento Local

O programa *blastn* é a versão do NCBI BLAST para a busca de similaridade de sequência de nucleotídeos.

Na saída ilustrada na Figura 13, correspondente a sétima etapa do *pipeline*, o usuário pode clicar no *link* BLAST de qualquer uma das sequências disponíveis. Essa ação desencadeará um novo *job* utilizando os parâmetros descritos na Tabela 4.

Dentre os recursos está uma lista de parâmetros obrigatórios de requisição para o BioServices do NCBI BLAST, apresentados na Tabela 4 e exemplificados no excerto de código a seguir:

```
@app.route('/blast/<int:count>')
def blast(count):
```

Tabela 4 – Detalhes dos parâmetros obrigatórios do NCBI BLAST

Parâmetro	Descrição	Valor
program	programa BLAST utilizado na execução da pesquisa	*blastn*
sequence	sequência de consulta	sequência no formato *fasta*
database	lista de nomes de bases de dados para pesquisa	*nt* (nucleotídeos)

```
print("blast")
try:
    if 'job_id' in session:
        print("session")
        job_id = session['job_id']
        fasta_file = USER_FILE_PATH + job_id + '.fasta'
        tab_out = USER_FILE_PATH + job_id
        hmm_result_list = Hmmer.process_result
                (fasta_file, tab_out, 10)
        print(hmm_result_list[count].sequence)
        url = Blast.qblast('blastn', 'nt',
                hmm_result_list[count].sequence)
    else:
        url = url_for('index')  #To do: error messsage
except Exception as e:
    print(e)
    url = url_for('index')#To do: error messsage
return redirect(url)
```

O BLAST lista os *hits* iniciais com a melhor correspondência (maior similaridade), como ilustrado na Figura 16. O primeiro elemento na lista, em azul, é a sequência alvo. Além dessa forma gráfica resumida, outras opções de saída estão disponíveis ao usuário, como a descrição de todas as sequências que produzem alinhamentos significativo, contendo o número de acesso (*Accession*), que é um código único que identifica uma sequência na base dados do NCBI, ilustrada pela Figura 17.

A terceira seção da saída BLAST, ilustrada na Figura 18, apresenta os alinhamentos que correspondem a todas as sequências na base de dados do NCBI que apresentaram algum significado por adesão à sequência fornecida como entrada pela ferramenta ao BLAST.

O BLAST também disponibiliza ao usuário outros relatórios, que podem ser acessados à partir do *link* com os resultados. As informações adicionais incluem o resumo da pesquisa, o relatório da toxina (linhagem), os resultados da árvore de distância, entre

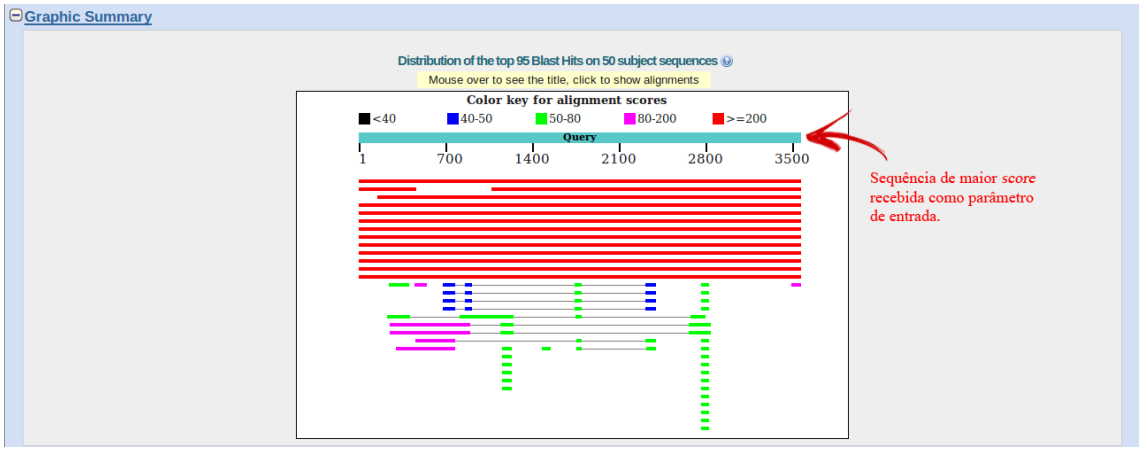


Figura 16 – Gráfico resumido do alinhamento de sequências.

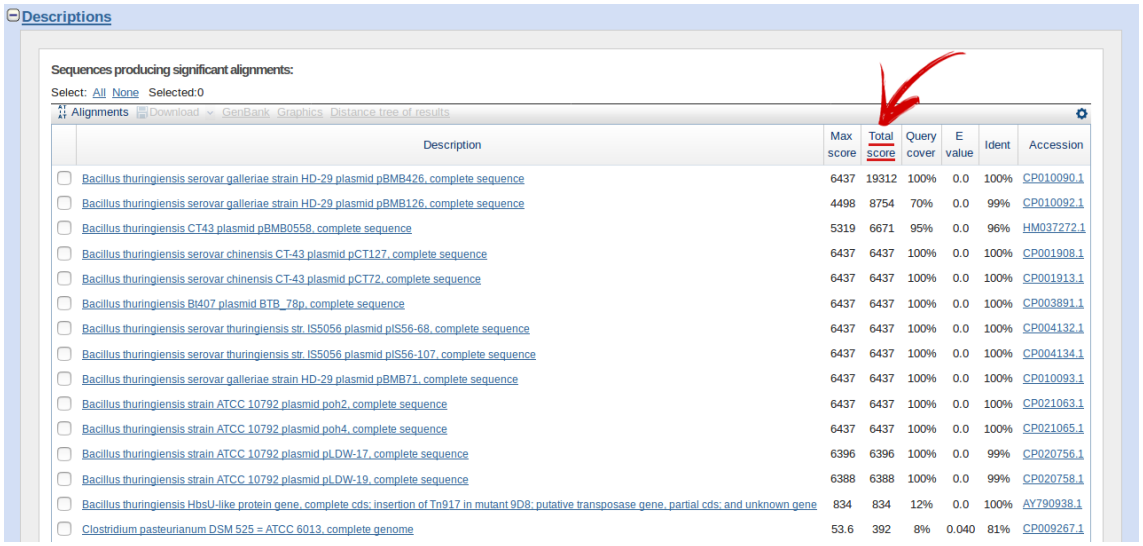


Figura 17 – Descrição das sequências mais significativas que foram alinhadas.

outros.

4.4 Considerações Gerais

A ferramenta foi pensada de forma que integrasse várias etapas das atividades que o usuário teria de fazer separadamente no dia a dia, proporcionando facilidade e agilidade ao seu trabalho e na obtenção das respostas. A ferramenta apresenta uma única entrada de dados, e a partir dela se desencadeia todo o processo de classificação e categorização.

O processo de classificação considera as sequências recebidas como entrada e o perfil HMM, buscando por intervalos de nucleotídeos que indiquem uma parte conservada de famílias de genes *cry*, apresentando essas sequências relevantes em ordem com base nos alinhamentos e pontuações (*score*) obtidas.

A categorização busca aprimorar as informações das sequências previamente clas-

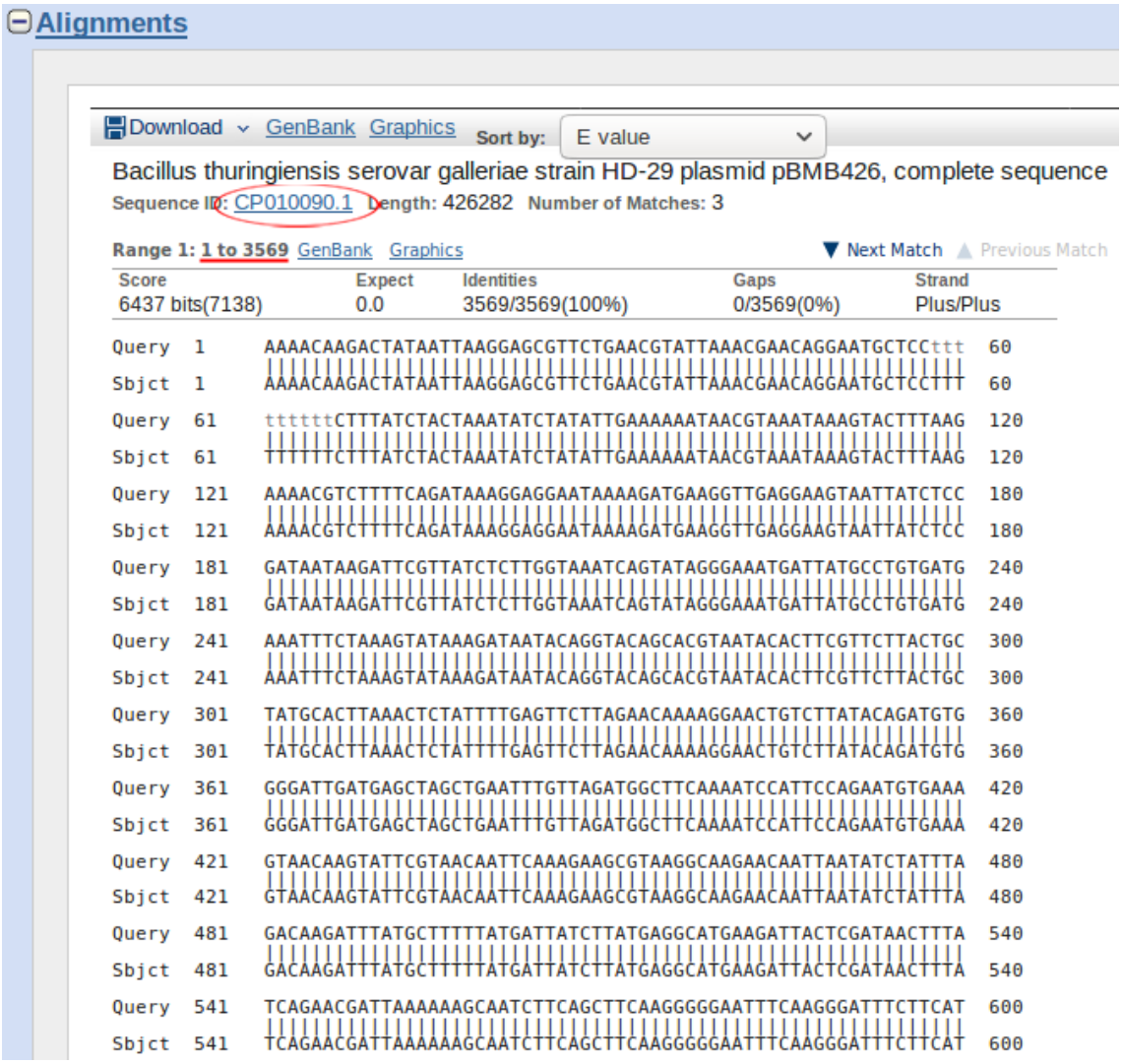


Figura 18 – Alinhamento das sequências mais significativas.

sificadas. Basta que o usuário clique sobre o *link* do programa na linha da sequência escolhida. O programa solicitado inicia um *job* que busca informações em base de dados públicas, gerando como saída os alinhamentos das sequências encontradas com sequências já publicadas e reconstruindo sua árvore filogenética, capaz de inferir a homologia necessária.

A ferramenta consegue, assim, alcançar o seu objetivo, que é o processo de receber e explorar grandes quantidades de nucleotídeos à procura de padrões consistentes para detectar relacionamentos sistemáticos entre as sequências obtidas e as sequências de perfil, detectando novos subconjuntos que representem possíveis genes *cry*.

5 RESULTADOS E DISCUSSÕES

Os testes foram realizados a partir de um arquivo no formato *FASTA* contendo o genoma de um isolado brasileiro da linhagem *B. thuringiensis* BR145, que apresenta atividade entomopatogênica contra *Helicoverpa armigera* e *Chrysodeixis includens*, importantes pragas agrícolas (RICIETO et al., 2012; RICIETTO, 2017).

Análises no genoma *B. thuringiensis* BR145 revelou os genes inseticidas *cry1Aa*, *cry1Ab*, *cry1Ac*, *cry1Ia* e *cry2Ab*, além dos genes *cyt1* e *vip3Aa* que não são alvos desse estudo (RICIETTO, 2017).

O DNA do genoma gerou um total de 3.042.174 *reads* de alta qualidade. O genoma consiste em 235 *contigs* com tamanho superior a 1000 bp, com um tamanho total de 6.350.733 bp (RICIETTO, 2017).

Uma análise dos *congtis* do BR145 utilizando o BLASTn contra a base de dados não redundante do NCBI identificou que o parente mais próximo é o *B. thuringiensis* var. *kurstaki* (RICIETTO, 2017).

A partir dessas informações foi realizado três tipos análises diferentes com o intuito de obter resultados da ferramenta DisCryGenes que comprovem a sua efetividade em detectar e encontrar novos genes *cry*:

1. Foi analisado os *contigs* do BR145 contra a totalidade da base curada de genes *cry*.
2. Foi analisado os *contigs* do BR145 contra 80% dos genes contidos na base de dados curada. Nessa etapa todas os grupos foram mantidos, mas 140 sequências foram retiradas da base de dados.
3. Foi analisado os *contigs* do BR145 contra 80% dos genes contidos na base de dados curada. A diferença para o segundo teste foi a retirada de três dos cinco grupos presentes nos resultados dos testes anteriores.

As seções seguintes abordam as análises e as discussões sobre os resultados encontrados. O Apêndice A ilustra algumas tabelas que ajudam a compreender os resultados das análises realizadas.

5.1 Análise Geral

A primeira etapa da execução do *pipeline* passa pelo HMMER, que executa o programa *nhmmscan* descrito na subseção 3.1.3 e gera uma tabela ilustrada pela Figura 27

na seção A.1 exibe parcialmente as 60 primeiras sequências de um total de 2600 linhas rotuladas pelo HMMER. O tempo de execução calculado pela ferramenta e armazenado no arquivo de configuração foi de 10 minutos e 53 segundos.

Após a primeira etapa da execução, a saída produzida pelo *nhmmscan* é filtrada através do *script filter-recode* que utiliza o método *merge* do BEDTools descrito na subseção 3.2.2 para combinar intervalos contíguos. O *script* gera duas saídas na forma tabular para o usuário analisar. A primeira tabela apresentada pela Figura 19 indica a localização de determinado gene no cromossomo. A tabela apresenta 3 colunas, que correspondem a *query name*, *alifrom* e *ali to* da saída do método *nhmmscan* do HMMER.

1	NODE_110_length_12556_cov_314.781	10454	11260
2	NODE_118_length_10549_cov_579.939	287	2187
3	NODE_118_length_10549_cov_579.939	7361	9577
4	NODE_118_length_10549_cov_579.939	10062	10549
5	NODE_11_length_133492_cov_140.468	109214	109417
6	NODE_124_length_8888_cov_128.481	5117	5304
7	NODE_166_length_3504_cov_159.262	2806	2904
8	NODE_172_length_3262_cov_120.933	2995	3168
9	NODE_186_length_2444_cov_1318.21	1	2444
10	NODE_18_length_108561_cov_132.893	17405	17503
11	NODE_200_length_1638_cov_2092.64	1015	1638
12	NODE_22_length_93611_cov_171.592	24508	24624
13	NODE_234_length_1013_cov_766.226	1	1013
14	NODE_235_length_1010_cov_839.119	1	1010
15	NODE_250_length_872_cov_1616.63	1	872
16	NODE_256_length_807_cov_903.879	1	506
17	NODE_25_length_82350_cov_196.987	35045	35100
18	NODE_262_length_735_cov_883.235	1	735
19	NODE_350_length_471_cov_1.02326	108	259
20	NODE_358_length_466_cov_432.563	22	231
21	NODE_358_length_466_cov_432.563	309	449
22	NODE_37_length_48629_cov_194.03	16674	16776
23	NODE_40_length_43792_cov_181.3	15419	15532
24	NODE_4_length_185155_cov_190.951	141452	141519
25	NODE_59_length_27247_cov_379.006	22373	22847
26	NODE_606_length_380_cov_4156.68	1	380
27	NODE_61_length_25976_cov_143.362	3164	3357
28	NODE_640_length_245_cov_986.542	1	245
29	NODE_655_length_158_cov_439.774	17	143
30	NODE_657_length_154_cov_44.0741	20	140
31	NODE_659_length_133_cov_839.833	24	129
32	NODE_67_length_23434_cov_215.945	23260	23325
33	NODE_71_length_22379_cov_167.84	13063	13134
34	NODE_72_length_22303_cov_155.942	21264	21333
35	NODE_7_length_146872_cov_132.479	9162	9290
36	NODE_7_length_146872_cov_132.479	51390	51506
37	NODE_84_length_19137_cov_360.453	7708	7853
38	NODE_84_length_19137_cov_360.453	10288	12188
39	NODE_87_length_18294_cov_409.757	1434	1632
40	NODE_87_length_18294_cov_409.757	15178	15719
41	NODE_94_length_15414_cov_99.2846	6355	6662
42	NODE_95_length_15200_cov_205.991	6134	6244
43	NODE_96_length_15085_cov_216.286	12514	12611

Figura 19 – *Locus* gênico: local do cromossomo ocupado por um gene.

Na seção A.1 do Apêndice A, a segunda tabela resultante do *script*, ilustrada na Figura 28, além das informações da localização do gene apresenta outras mais gerais da saída do HMMER, como *E-value*, *strand*, *target name*, *score* e *bias*.

Quando foi aplicado o método *nhmmscan* do HMMER obtivemos 2600 sequências candidatas a genes *cry*. Ao aplicarmos o *script filter-recode* à saída do HMMER, o método *merge* foi capaz de reduzir para 43 sequências candidatas a genes *cry*. Comparamos os resultados da tese de doutorado da Dra. Ana Paula Scaramal Ricietto (RICIETTO, 2017) com os resultados obtidos a partir da execução da ferramenta. A Tabela 5 apresenta os resultados obtidos da tese de doutorado (RICIETTO, 2017), enquanto a Tabela 6 apresenta os resultados obtidos a partir da ferramenta desenvolvida nessa dissertação.

Tabela 5 – Resultados da tese de doutorado da Dra. Ana Paula Scaramal Ricietto

Id	**Início**	**Fim**	**Nome**	**E-value**	**Bit Score**	**Bias**
NODE_130_length_10549_cov_312.223	7361	9577	Cry1Ia	0	2498.7	115.5
NODE_130_length_10549_cov_312.223	10062	10549	Cry1Ac	2.8e-161	535.7	18.1
NODE_206_length_2444_cov_691.284	1	2444	Cry1Ac	0	2864.7	92.0
NODE_224_length_1638_cov_1119.85	1015	1638	Cry1Aa	7.8e-210	694.0	31.6
NODE_257_length_1013_cov_371.731	1	1013	Cry1Ab	0	1153.5	42.4
NODE_258_length_1010_cov_445.196	1	1010	Cry1Aa	0	1175.7	33.3
NODE_269_length_872_cov_844.133	1	872	Cry1Aa	2.2e-301	996.8	44.8
NODE_276_length_807_cov_491.524	1	506	Cry1Ab	2.4e-163	538.7	17.5
NODE_285_length_735_cov_452.327	1	735	Cry1Ab	1.6e-255	844.3	26.7
NODE_506_length_380_cov_2236.51	1	380	Cry1Ac	4.0e-130	427.5	8.1
NODE_539_length_245_cov_526.28	1	245	Cry1Aa	1.9e-79	258.7	10.8
NODE_88_length_19137_cov_193.565	10288	12188	Cry2Ab	0	2172.2	131.0

Tabela 6 – Resultados da Ferramenta para Descoberta de Genes *cry*

Id	**Início**	**Fim**	**Nome**	**E-value**	**Bit Score**	**Bias**
NODE_118_length_10549_cov_579.939	7361	9577	Cry1Ia	0	2496.6	115.5
NODE_118_length_10549_cov_579.939	10062	10549	Cry1Ac	3.2e-161	535.7	18.1
NODE_186_length_2444_cov_1318.21	1	2444	Cry1Ac	0	2864.7	92.0
NODE_200_length_1638_cov_2092.64	1015	1638	Cry1Aa	9e-210	694.0	31.6
NODE_234_length_1013_cov_766.226	1	1013	Cry1Ab	0	1153.5	42.4
NODE_235_length_1010_cov_839.119	1	1010	Cry1Aa	0	1175.7	33.3
NODE_250_length_872_cov_1616.63	1	872	Cry1Aa	2.5e-301	996.8	44.8
NODE_256_length_807_cov_903.879	1	506	Cry1Ab	2.7e-163	538.7	17.5
NODE_262_length_735_cov_883.235	1	735	Cry1Ab	1.8e-255	844.3	26.7
NODE_606_length_380_cov_4156.68	1	380	Cry1Ac	4.5e-130	427.5	8.1
NODE_640_length_245_cov_986.542	1	245	Cry1Aa	2.2e-79	258.7	10.8
NODE_84_length_19137_cov_360.453	10288	12188	Cry2Ab	0	2172.2	131.0

Comparando as duas tabelas é possível verificar que foram encontrados os mesmos genes *cry* em ambos os trabalhos, com o mesmo *locus*. No entanto, é possível perceber que os resultados apresentam alguma variação no *E-values*, que é o parâmetro de confiança dos alinhamentos. O *E-value* indica o número de alinhamentos que seriam esperados apresentando valores de *score* iguais ou melhores que o encontrado por acaso, dado o tamanho do base de dados. Os *E-values* do HMMER tendem a ser precisos, e *E-values* de 0,1 ou inferiores são, em geral, acertos muito significativos (FINN; CLEMENTS; EDDY, 2011). A base de dados de perfis *hmm* é de tamanho variável e composto de sequências

homólogas. Como os valores de *E-value* tendem sempre a zero, podemos afirmar que os acertos são significativos.

O *bit score* é outro indicador estatístico para medir a similaridade de uma sequência independente do comprimento da sequência da consulta e do tamanho da base de dados. Assim, quanto maior for o *bit score*, mais significativo é o acerto (FINN; CLEMENTS; EDDY, 2011). O *bit score* fornece um indicador estatístico constante para busca em diferentes bases de dados de diferentes tamanhos ou para buscar na mesma base de dados em momentos diferentes à medida que a base de dados aumenta. Sabendo disso, é possível perceber e conferir que os valores do *bit score* nos dois resultados são praticamente iguais, mantendo a significância dos acertos, mesmo sendo executados em momentos distintos.

A coluna com o número *bias* é um termo de correção para a composição de uma sequência tendenciosa aplicada a sequência de *bit score*. A possibilidade de ajuste da correção de tendência pode permitir a preservação de uma sequência não homóloga com alta pontuação ou causar a perda de uma sequência homóloga (EDDY, 1992). Em ambos os resultados foram utilizados os mesmos valores *default* para o valor *bias*.

5.1.1 Resultados da Categorização

As ferramentas de categorização auxiliam na continuidade do trabalho do usuário para classificar uma determinada sequência alvo. Nessa etapa da execução a ferramenta já devolveu o nome e a localização das sequências. Das 43 sequências resultantes, escolhemos a terceira sequência na Tabela 6, que corresponde a nona sequência listada na Figura 19, para analisar as informações disponíveis sobre essa sequência nas bases de dados pública.

A partir do identificador da sequência (*query name*), o intervalo (*alifrom* e *ali to*) é selecionado e a sequência é armazenada como um arquivo no formato FASTA. Quando essa sequência for executada no MUSCLE ela é incluída as sequências que fazem parte do *target name*, que nesse caso é Cry1Ac. Já o BLAST utiliza a sequência individualmente contra a sua base de nucleotídeos.

A Figura 20 ilustra os resultados para o *job* do MUSCLE. Apresenta os alinhamentos múltiplos das sequências e a árvore filogenética a partir do método *Neighbour-joining* sem correções de distância.

Como visto na subseção 3.3.2, a reconstrução da árvore filogenética utiliza a matriz de similaridade. A Figura 21 ilustra a matriz de similaridade, através dela é possível apontar as sequências na base de dados com maior identidade ou homologia com a sequência alvo. A nossa sequência alvo é a 19ª sequência listada na matriz, tanto na vertical quanto na horizontal. Em relação a primeira sequência da matriz, por exemplo, identificada como Cry1Ac35, o percentual de identidade é de 99,67%. Essa análise pode prosseguir para as outras sequências listadas na matriz identidade.

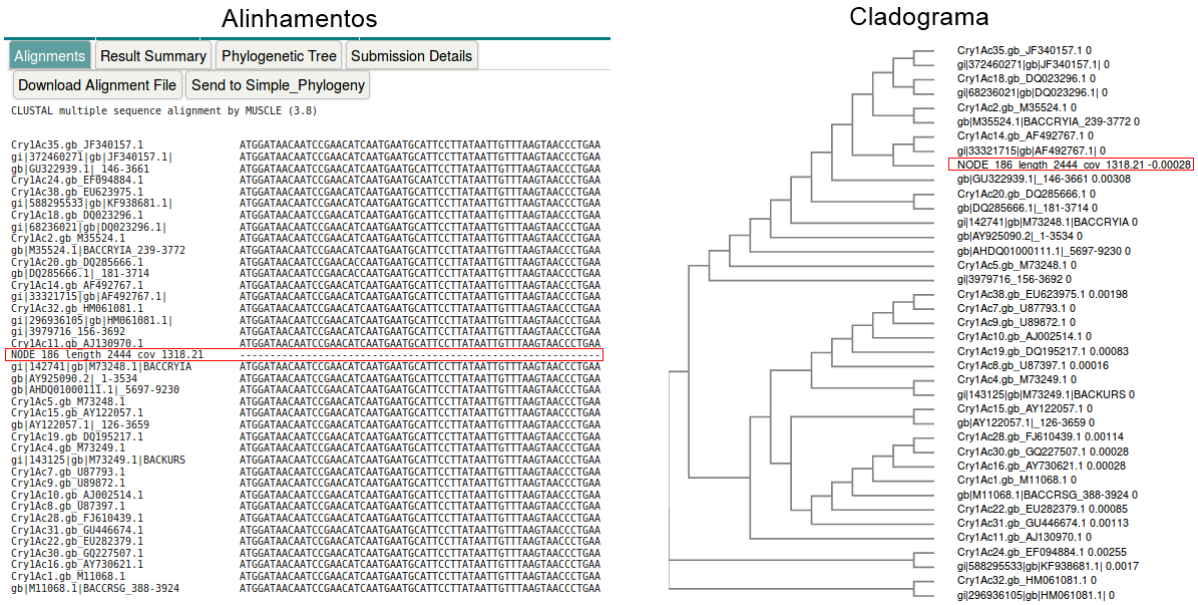


Figura 20 – Informações do MUSCLE sobre os alinhamentos e ancestralidade.

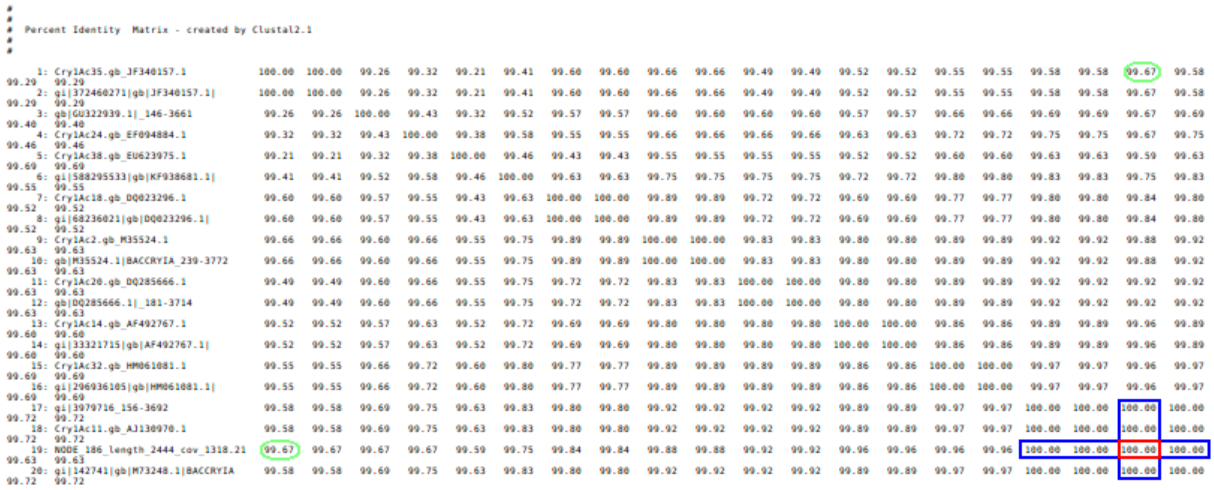


Figura 21 – Informações do MUSCLE sobre a matriz identidade.

Observando a Figura 21 notamos os valores com maior percentual de identidade contornados em azul, e a sequência alvo, no centro, contornada em vermelho.

A estrutura dos resultados do BLAST envolve *Graphic Summary*, *Descriptions* e *Alignments*. Por intermédio do *Graphic Summary*, ilustrado na Figura 22, é possível que o usuário passe o mouse sobre as linhas e veja o nome das sequências submetidas. O usuário também pode optar por clicar sobre uma linha para obter mais informações sobre a sequência. As cores das linhas podem variar de acordo com as pontuações obtidas nos alinhamentos.

A Figura 23 permite que o usuário analise quão significativos são os alinhamentos das sequências mantidas na base de dados do NCBI com a sequência alvo, observando e ordenando as sequências pelo *Max score*, a pontuação mais alta, pelo *Total score*, as

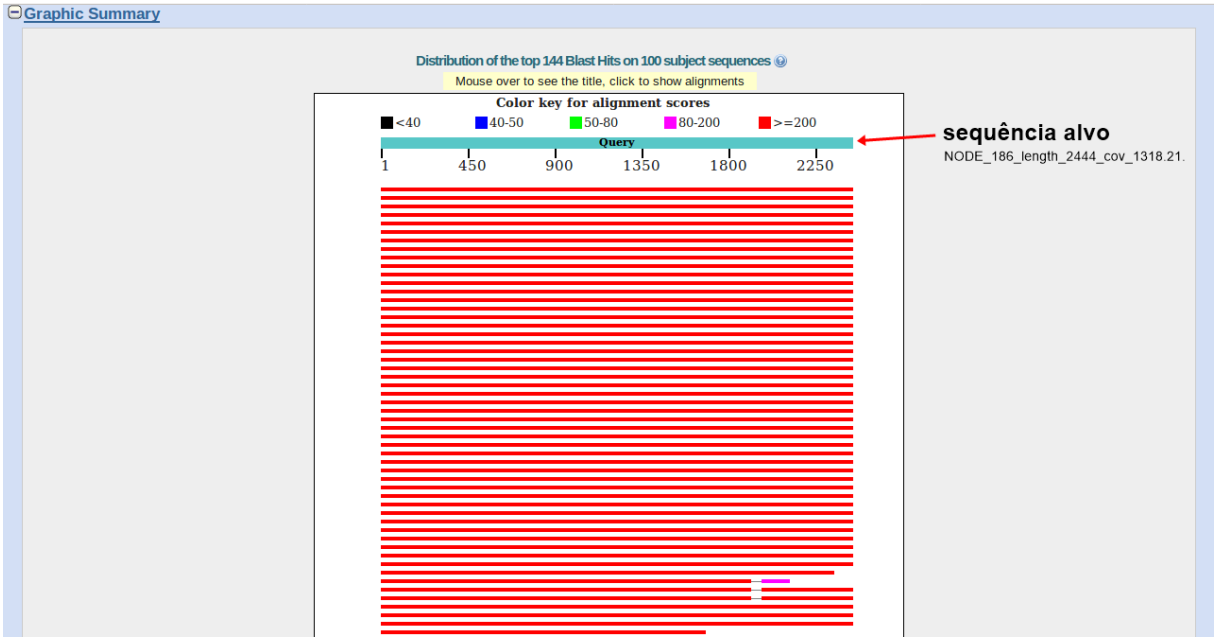
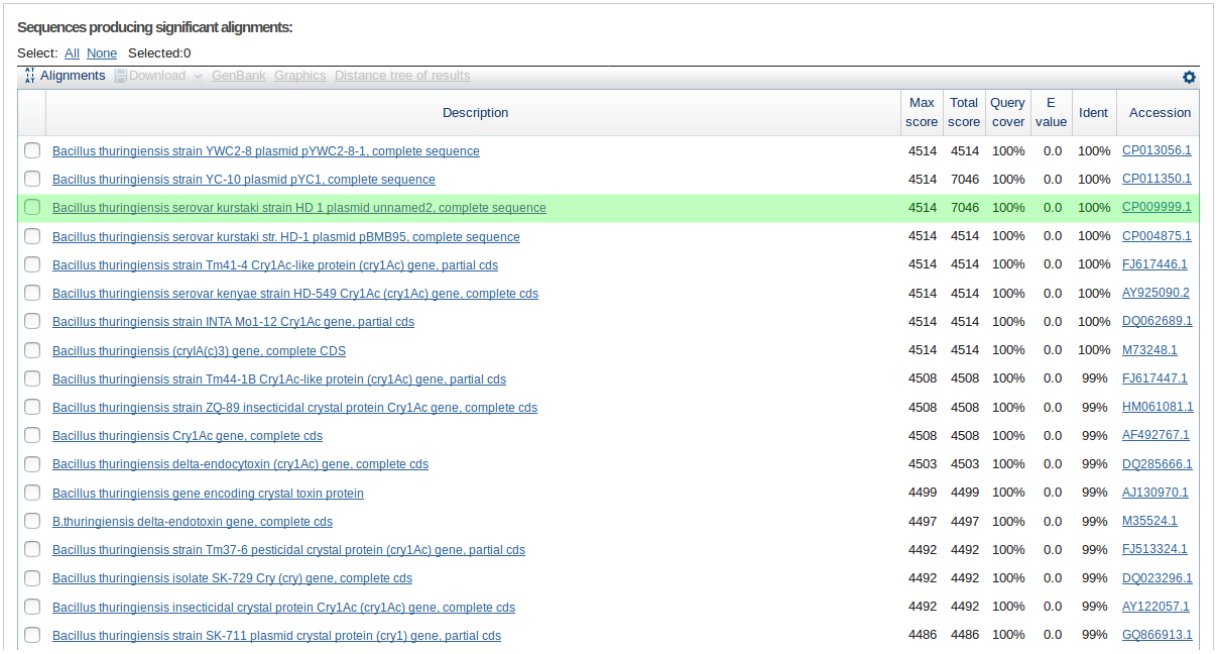


Figura 22 – Informações resumidasdo BLAST sobre os alinhamentos mais significativos.

pontuações totais de alinhamentos, *Query cover*, o percentual de cobertura de alinhamento da sequência consultada pela sequência na base de dados, *E-value* (o mais baixo) e *Ident*, o percentual mais elevado de identidade de todas os alinhamentos.



Sequences producing significant alignments:

Select: All None Selected:0

Alignments Download GenBank Graphics Distance tree of results

Description	Max score	Total score	Query cover	E value	Ident	Accession
Bacillus thuringiensis strain YWC2-8 plasmid pYWC2-8-1, complete sequence	4514	4514	100%	0.0	100%	CP013056.1
Bacillus thuringiensis strain YC-10 plasmid pYC1, complete sequence	4514	7046	100%	0.0	100%	CP011350.1
Bacillus thuringiensis serovar kurstaki strain HD 1 plasmid unnamed2, complete sequence	4514	7046	100%	0.0	100%	CP009999.1
Bacillus thuringiensis serovar kurstaki str. HD-1 plasmid pBMB95, complete sequence	4514	4514	100%	0.0	100%	CP004875.1
Bacillus thuringiensis strain Tm41-4 Cry1Ac-like protein (cry1Ac) gene, partial cds	4514	4514	100%	0.0	100%	FJ617446.1
Bacillus thuringiensis serovar kenyae strain HD-549 Cry1Ac (cry1Ac) gene, complete cds	4514	4514	100%	0.0	100%	AY925090.2
Bacillus thuringiensis strain INTA Mo1-12 Cry1Ac gene, partial cds	4514	4514	100%	0.0	100%	DQ062689.1
Bacillus thuringiensis (crylA(c)3) gene, complete CDS	4514	4514	100%	0.0	100%	M73248.1
Bacillus thuringiensis strain Tm44-1B Cry1Ac-like protein (cry1Ac) gene, partial cds	4508	4508	100%	0.0	99%	FJ617447.1
Bacillus thuringiensis strain ZQ-89 insecticidal crystal protein Cry1Ac gene, complete cds	4508	4508	100%	0.0	99%	HM061081.1
Bacillus thuringiensis Cry1Ac gene, complete cds	4508	4508	100%	0.0	99%	AF492767.1
Bacillus thuringiensis delta-endocytoxin (cry1Ac) gene, complete cds	4503	4503	100%	0.0	99%	DQ285666.1
Bacillus thuringiensis gene encoding crystal toxin protein	4499	4499	100%	0.0	99%	AJ130970.1
B.thuringiensis delta-endotoxin gene, complete cds	4497	4497	100%	0.0	99%	M35524.1
Bacillus thuringiensis strain Tm37-6 pesticidal crystal protein (cry1Ac) gene, partial cds	4492	4492	100%	0.0	99%	FJ513324.1
Bacillus thuringiensis isolate SK-729 Cry (cry) gene, complete cds	4492	4492	100%	0.0	99%	DQ023296.1
Bacillus thuringiensis insecticidal crystal protein Cry1Ac (cry1Ac) gene, complete cds	4492	4492	100%	0.0	99%	AY122057.1
Bacillus thuringiensis strain SK-711 plasmid crystal protein (cry1) gene, partial cds	4486	4486	100%	0.0	99%	GQ866913.1

Figura 23 – Informações discritivas dos alinhamentos do BLAST.

Para finalizar a análise, o usuário pode selecionar as sequências na tabela *Descriptions* para ver os alinhamentos. Como sabemos que o parente mais próximo é o *Bacillus thuringiensis* var *kurstaki*, selecionamentos a terceira sequência para ilustrar os alinhamentos, como exemplificado na Figura 24.

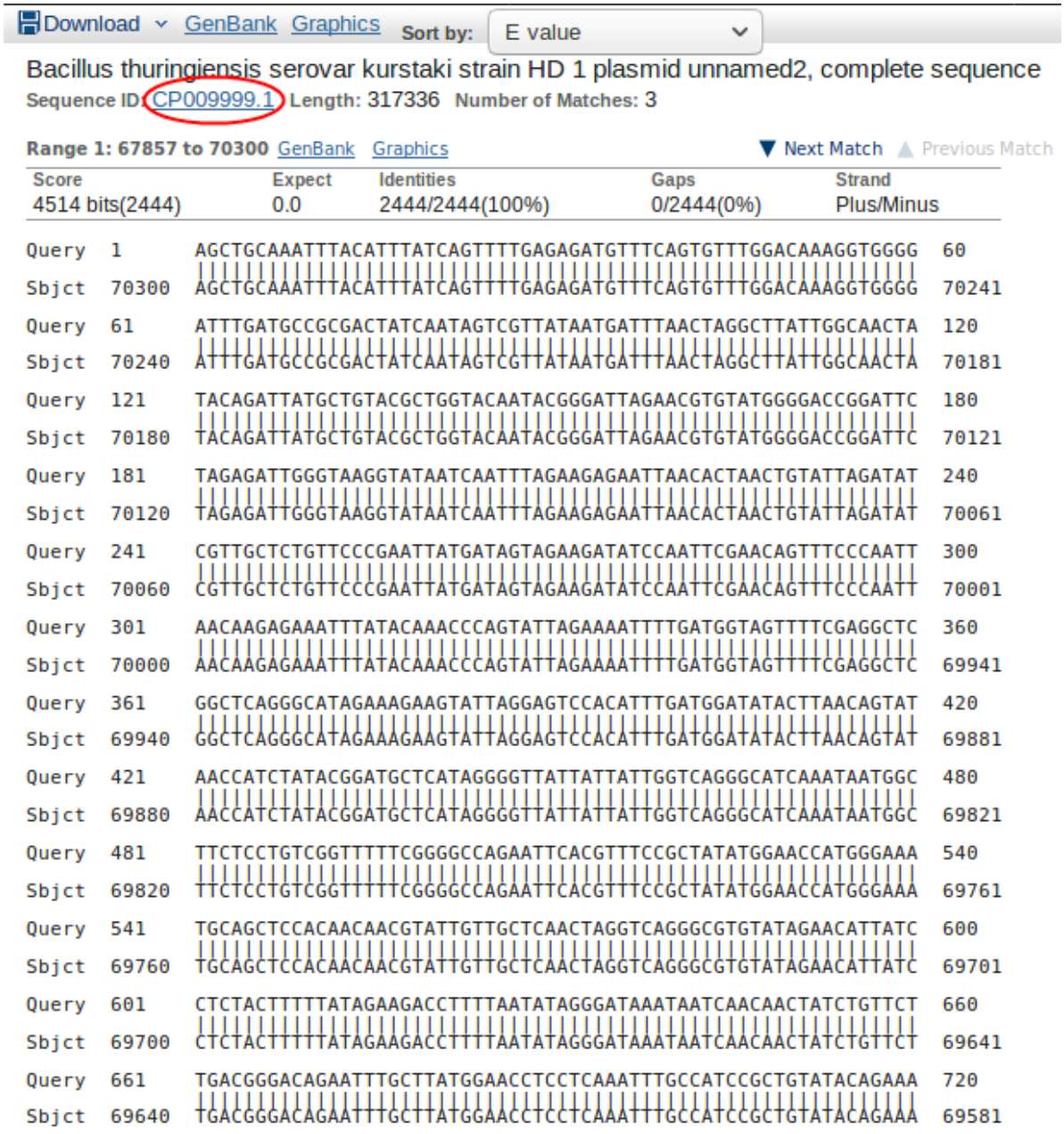


Figura 24 – Alinhamento da sequência alvo e a sequência da base de dados do NCBI.

As famílias gênicas que codificam a proteína Cry são muito parecidas, o que dificulta tomar uma decisão quanto a sua correta categorização. Justamente por isso torna-se necessário utilizar diferentes estratégias para se chegar a um resultado razoável. O programa faz a localização no genoma dos genes *cry*, compara com a base de dados e dá uma indicação muito próxima sobre a qual família pertence. Mas, realmente, é recomendável que o usuário pegue estas sequências e faça outras comparações, utilizando ferramentas como o BLAST ou o MUSCLE, entre outras.

5.2 Análise de 80% das Sequências

Para o segundo teste foi retirado 20% das sequências de todas os grupos de genes *cry* anotados na base de dados. Foram retiradas 140 sequências da base de dados, respeitando-se, no entanto, que todos os grupos estivessem representados na base de dados. Em cada família, a cada 3 sequências uma foi retirada da base de dados original.

Após retirar as sequências houve a necessidade de se construir os novos perfis *hmm* de cada grupo. Utilizamos o método *hmmbuild* descrito na subseção 3.1.2. Com os perfis de todas as famílias criados foi necessário concatená-los em um único arquivo que representa a base de dados de perfis HMM. Com um único arquivo contendo todos os perfis foi executado o método *hmmpress* para compactar e gerar os os arquivos binários como descreve a subseção 3.1.3. Essa etapa, embora não esteja automatizada na ferramenta, foi descrita na subseção 4.3.1 referente ao *pipeline*.

Construído o modelo HMM passamos para a busca de sequências de nucleotídeos utilizando os *contigs* do BR145 contra o perfil de nucleotídeos da nova base de dados. Após executar o método *nhmmscan* foi obtido a tabela ilustrada parcialmente na Figura 29, na seção A.2 do Apêndice A, que retornou 2633 sequências candidatas a genes que codifiquem a proteína Cry.

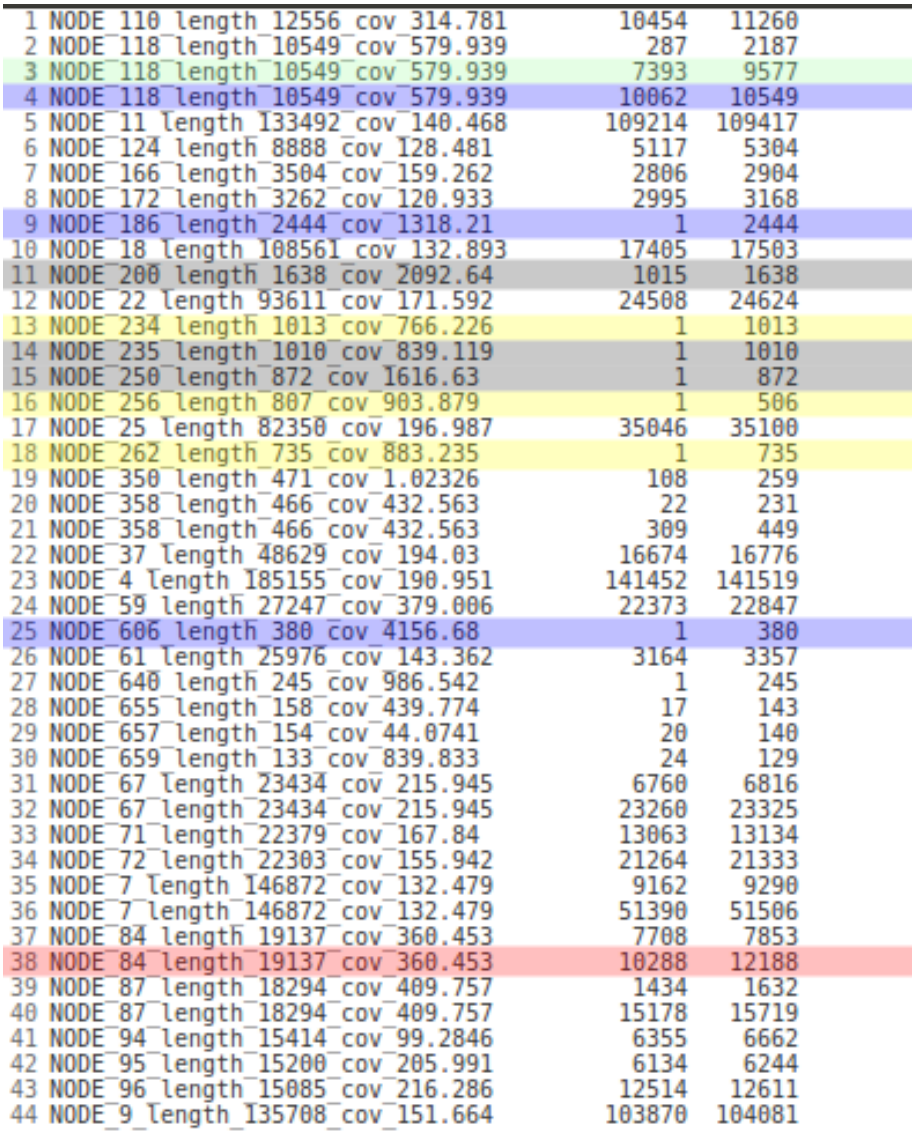
Para filtrar o resultado foi aplicado o *script filter-recode*, que reduziu o resultado para 44 sequências candidatas, como ilusta a Figura 25 que mostra a localização gênica e a Figura 30 (seção A.2) que apresenta informações sobre os genes *cry* candidatos.

Analisando a Tabela 7 é possível verificar uma divergência no *locus* do primeiro gene *cry* encontrado nesse segundo teste em relação ao trabalho original apresentado pela Dra. Ana Paula Scaramal Ricietto indicado na Tabela 5. O segundo teste apresenta uma diferença de 32 bp à frente com relação à posição inicial do teste original. Outro ponto que se altera são os valores nas colunas *E-value* e *bit score*.

Como o *bit score* é uma pontuação que compara a probabilidade do perfil HMM com a probabilidade de uma hipótese nula (EDDY, 1998), a divergência é resultado das sequências que foram excluídas para a realização desse segundo teste.

5.3 Análise sem Grupos de Famílias Conhecidos

O terceiro teste considerou os mesmos *contigs* do BR145 e utilizou a base de dados com 20% das sequências originais excluídas, sendo que as sequências retiradas faziam parte dos resultados conhecidos previamente. No primeiro teste com a base de dados completa, por exemplo, foram encontradas as famílias Cry1Aa, Cry1Ab, Cry1Ac, Cry1Ia e Cry2Ab nos resultados. Para esse terceiro teste foram retiradas todas as sequências das famílias Cry1Aa, Cry1Ab, Cry1Ac e Cry1Ba, também totalizando 140 sequências a



1	NODE_110_length_12556_cov_314.781	10454	11260
2	NODE_118_length_10549_cov_579.939	287	2187
3	NODE_118_length_10549_cov_579.939	7393	9577
4	NODE_118_length_10549_cov_579.939	10062	10549
5	NODE_11_length_133492_cov_140.468	109214	109417
6	NODE_124_length_8888_cov_128.481	5117	5304
7	NODE_166_length_3504_cov_159.262	2806	2904
8	NODE_172_length_3262_cov_120.933	2995	3168
9	NODE_186_length_2444_cov_1318.21	1	2444
10	NODE_18_length_108561_cov_132.893	17405	17503
11	NODE_200_length_1638_cov_2092.64	1015	1638
12	NODE_22_length_93611_cov_171.592	24508	24624
13	NODE_234_length_1013_cov_766.226	1	1013
14	NODE_235_length_1010_cov_839.119	1	1010
15	NODE_250_length_872_cov_1616.63	1	872
16	NODE_256_length_807_cov_903.879	1	506
17	NODE_25_length_82350_cov_196.987	35046	35100
18	NODE_262_length_735_cov_883.235	1	735
19	NODE_350_length_471_cov_1.02326	108	259
20	NODE_358_length_466_cov_432.563	22	231
21	NODE_358_length_466_cov_432.563	309	449
22	NODE_37_length_48629_cov_194.03	16674	16776
23	NODE_4_length_185155_cov_190.951	141452	141519
24	NODE_59_length_27247_cov_379.006	22373	22847
25	NODE_606_length_380_cov_4156.68	1	380
26	NODE_61_length_25976_cov_143.362	3164	3357
27	NODE_640_length_245_cov_986.542	1	245
28	NODE_655_length_158_cov_439.774	17	143
29	NODE_657_length_154_cov_44.0741	20	140
30	NODE_659_length_133_cov_839.833	24	129
31	NODE_67_length_23434_cov_215.945	6760	6816
32	NODE_67_length_23434_cov_215.945	23260	23325
33	NODE_71_length_22379_cov_167.84	13063	13134
34	NODE_72_length_22303_cov_155.942	21264	21333
35	NODE_7_length_146872_cov_132.479	9162	9290
36	NODE_7_length_146872_cov_132.479	51390	51506
37	NODE_84_length_19137_cov_360.453	7708	7853
38	NODE_84_length_19137_cov_360.453	10288	12188
39	NODE_87_length_18294_cov_409.757	1434	1632
40	NODE_87_length_18294_cov_409.757	15178	15719
41	NODE_94_length_15414_cov_99.2846	6355	6662
42	NODE_95_length_15200_cov_205.991	6134	6244
43	NODE_96_length_15085_cov_216.286	12514	12611
44	NODE_9_length_135708_cov_151.664	103870	104081

Figura 25 – *Locus* gênico: a localização dos genes utilizando 80% da base de dados.

Tabela 7 – Resultados da Ferramenta para Descoberta de Genes *cry* utilizando 80% das sequências na base de dados

Id	**Início**	**Fim**	**Nome**	**E-value**	**Bit Score**	**Bias**
NODE_118_length_10549_cov_579.939	7393	9577	Cry1Ia	0	2504.2	115.5
NODE_118_length_10549_cov_579.939	10062	10549	Cry1Ac	8.1e-162	537.4	18.1
NODE_186_length_2444_cov_1318.21	1	2444	Cry1Ac	0	2863.2	92.0
NODE_200_length_1638_cov_2092.64	1015	1638	Cry1Aa	5.8e-208	687.8	31.6
NODE_234_length_1013_cov_766.226	1	1013	Cry1Ab	0	1142.7	42.4
NODE_235_length_1010_cov_839.119	1	1010	Cry1Aa	0	1163.6	33.3
NODE_250_length_872_cov_1616.63	1	872	Cry1Ab	1.1e-298	987.9	44.8
NODE_256_length_807_cov_903.879	1	506	Cry1Ab	1.3e-165	546.3	17.6
NODE_262_length_735_cov_883.235	1	735	Cry1Ab	5.9e-258	852.5	26.7
NODE_606_length_380_cov_4156.68	1	380	Cry1Ab	4.5e-131	430.6	8.1
NODE_640_length_245_cov_986.542	1	245	Cry1Aa	1.6e-78	255.6	10.8
NODE_84_length_19137_cov_360.453	10288	12188	Cry2Ab	0	2170.2	131.0

menos do original, exatamente como no segundo teste, sendo que, dessas sequências que foram retiradas, três são de famílias inteiras de sequências que aparecem nos resultados das análises anteriores.

Após retirar essas sequências da base de dados foi construído os novos perfis *hmm* de cada grupo utilizando o método *hmmbuild*. Todos os perfis criados foram concatenados em um único arquivo que passou a representar a nova base de dados de perfis HMM. O arquivo contendo todos os perfis foi compactado pelo método *hmmpress*, gerando os arquivos binários necessários para a execução da busca de nucletídeos em perfis *hmm* com o método *nhmmscan*.

A execução do método *nhmmscan* resultou em 2564 linhas contendo sequências candidatas a genes *cry*, mostradas parcialmente na Figura 31 (seção A.3 do Apêndice A).

A segunda etapa do *pipeline* consiste na execução do *script filter-recode* que limitou em 43 sequências candidatas, como é possível verificar na Figura 26, que exibe o identificador da sequência e a sua posição inicial e final na sequência alvo.

1	NODE_110_length_12556_cov_314.781	10454	11260
2	NODE_118_length_10549_cov_579.939	287	2187
3	NODE_118_length_10549_cov_579.939	7393	9577
4	NODE_118_length_10549_cov_579.939	10062	10549
5	NODE_11_length_133492_cov_140.468	109214	109417
6	NODE_124_length_8888_cov_128.481	5117	5304
7	NODE_166_length_3504_cov_159.262	2806	2904
8	NODE_172_length_3262_cov_120.933	2995	3168
9	NODE_186_length_2444_cov_1318.21	1	2444
10	NODE_18_length_108561_cov_132.893	17405	17503
11	NODE_200_length_1638_cov_2092.64	1015	1638
12	NODE_22_length_93611_cov_171.592	24508	24624
13	NODE_234_length_1013_cov_766.226	1	1013
14	NODE_235_length_1010_cov_839.119	1	1010
15	NODE_250_length_872_cov_1616.63	1	872
16	NODE_256_length_807_cov_903.879	1	506
17	NODE_25_length_82350_cov_196.987	35045	35100
18	NODE_262_length_735_cov_883.235	1	735
19	NODE_350_length_471_cov_1.02326	108	259
20	NODE_358_length_466_cov_432.563	22	231
21	NODE_358_length_466_cov_432.563	309	449
22	NODE_37_length_48629_cov_194.03	16674	16776
23	NODE_4_length_185155_cov_190.951	141452	141519
24	NODE_59_length_27247_cov_379.006	22373	22847
25	NODE_606_length_380_cov_4156.68	1	380
26	NODE_61_length_25976_cov_143.362	3164	3357
27	NODE_640_length_245_cov_986.542	1	245
28	NODE_655_length_158_cov_439.774	17	143
29	NODE_657_length_154_cov_44.0741	20	140
30	NODE_659_length_133_cov_839.833	24	129
31	NODE_67_length_23434_cov_215.945	23260	23325
32	NODE_71_length_22379_cov_167.84	13063	13134
33	NODE_72_length_22303_cov_155.942	21264	21333
34	NODE_7_length_146872_cov_132.479	9162	9290
35	NODE_7_length_146872_cov_132.479	51390	51506
36	NODE_84_length_19137_cov_360.453	7708	7853
37	NODE_84_length_19137_cov_360.453	10288	12188
38	NODE_87_length_18294_cov_409.757	1434	1632
39	NODE_87_length_18294_cov_409.757	15178	15719
40	NODE_94_length_15414_cov_99.2846	6355	6662
41	NODE_95_length_15200_cov_205.991	6134	6244
42	NODE_96_length_15085_cov_216.286	12514	12611
43	NODE_9_length_135708_cov_151.664	103870	104081

Figura 26 – *Locus* gênico: a localização dos genes retirando parte das famílias de genes *cry*.

Observando os resultados obtidos no primeiro teste, ilustrado na Figura 19 e os resultados obtidos no terceiro teste, ilustrado Figura 26, é possível verificar que, com exceção do primeiro *hit* que tem uma diferença de 32 bp à frente, todas as outras sequências apresentam exatamente as mesmas localizações na sequência alvo. Os detalhes dos resultados são ilustrados na Figura 32 na seção A.3 do Apêndice A.

Analisando a Tabela 8 é possível notar em relação ao teste original uma divergência nos dados estatísticos representados pelas colunas *E-value* e *Bit score*, além da coluna *Nome*, haja visto que três famílias finalistas foram retiradas da base. No entanto, a primeira sequência, além de divergir na localização inicial e em tamanho, também diverge parcialmente no nome da família, mesmo ela estando na base de dados. O correto seria apontar para a família Cry1Ia, porém a ferramenta acerta somente até o segundo ranque, no terceiro ranque ele troca *a* por *e*. Até o segundo ranque as sequências de mesma família apresentam até 78% de identidade, enquanto que no terceiro ranque chega a até 95% de identidade, como explicado na subseção 2.2.1.

Tabela 8 – Resultados da Ferramenta para Descoberta de Genes *cry* utilizando 80% das sequências na base de dados

Id	**Início**	**Fim**	**Nome**	**E-value**	**Bit Score**	**Bias**
NODE_118_length_10549_cov_579.939	7393	9577	Cry1Ie	0	2233.6	115.5
NODE_118_length_10549_cov_579.939	10062	10549	Cry1Fa	1.1e-145	484.1	18.1
NODE_186_length_2444_cov_1318.21	1	2444	Cry1Ah	0	2579.0	91.8
NODE_200_length_1638_cov_2092.64	1015	1638	Cry1Ai	4.5e-207	684.8	31.6
NODE_234_length_1013_cov_766.226	1	1013	Cry1Af	1.9e-300	994.0	41.5
NODE_235_length_1010_cov_839.119	1	1010	Cry1Ai	0	1072.2	32.5
NODE_250_length_872_cov_1616.63	1	872	Cry1A-like	2.4e-263	870.5	44.8
NODE_256_length_807_cov_903.879	1	506	Cry1Ai	8e-157	517.1	17.4
NODE_262_length_735_cov_883.235	1	735	Cry1Ca	5.9e-204	673.5	26.7
NODE_606_length_380_cov_4156.68	1	380	Cry1Ga	6.6e-120	393.6	7.8
NODE_640_length_245_cov_986.542	1	245	Cry1Ga	3.9e-75	244.4	10.8
NODE_84_length_19137_cov_360.453	10288	12188	Cry2Ab	0	2172.2	131.0

5.4 Discussões

Ao executarmos o primeiro teste descrito na seção 5.1, onde a análise considera todas as sequências conhecidas na base de dados, serviu para ratificar que a base de dados estava funcional e que sequências iguais ou com alto grau de identidade poderiam ser encontradas e classificadas corretamente.

Ao analisarmos os resultados obtidos no segundo teste, descrito na seção 5.2, onde foram retiradas 140 sequências das 699 contidas na base de dados original, porém, mantendo-se todas as famílias. A ferramenta foi capaz de identificar todas as sequências corretamente, tendo apenas discrepância de 32 bp na posição inicial do primeiro gene *cry* apontado como correto. Isso significa que os genes *cry* puderam ser localizados, mesmo

sem parte da base de dados, e se tivesse um gene *cry* desconhecido pertencente a mesma família presente na base de dados ele seria localizado.

No último teste também foram retiradas 140 sequências da base de dados original, porém, nesse caso as sequência retiradas representaram grupos inteiros de genes *cry* que apareciam como finalistas na primeira análise com a base de dados original. Como descrito na seção 5.3, as mesmas sequências foram encontradas, apontando a posição inicial e final corretamente, com exceção do primeiro gene *cry* detectado que continuou com uma variação na posição inicial de 32 bp à frente da posição original. Diante das análises realizadas e dos resultados obtidos foi possível perceber que mesmo sem o modelo para um grupo de nucleotídeos, eles foram localizados.

Imaginando uma nova proteína Cry, portanto, sem um modelo, a ferramenta seria capaz de identificá-la no genoma, mesmo que parcialmente. Para descrevê-la seria necessário estudá-la especificamente com a participação do especialista. Com isso, o objetivo da ferramenta de classificar e ou localizar novos genes de um grupo foi atingido.

6 CONSIDERAÇÕES FINAIS

Nesse trabalho foi realizado um estudo sobre *Bacillus thuringiensis*, mais especificamente sobre a sua capacidade de produzir proteína cristal que apresenta toxicidade contra diversas Ordens de insetos, tratado no Capítulo 2.

Paralelo ao estudo sobre Bt, foi realizado outro estudo envolvendo quatro ferramentas de bioinformática que foram utilizadas no trabalho para auxiliar na detecção e categorização de genes *cry*, e desenvolvido um ambiente *web* com capacidade de iniciar os serviços disponíveis por essas ferramentas de bioinformática. O trabalho envolve as seguintes funcionalidades e recursos:

1. A partir do software HMMER foi criado um perfil HMM de todas as sequências conhecidas de genes *cry*, obtidas a partir de uma base dados curada;
2. Por intermédio do ambiente *web* o usuário informa uma sequência de nucleotídeos, e é realizada uma busca por homologia nesse perfil HMM, utilizando os software HMMER e BEDTools, com a intenção de apresentar ao usuário um resultado sumarizado de todas as possíveis sequências de genes *cry* contidas na entrada que foi fornecida;
3. O usuário pode, então, escolher qualquer uma das sequências classificadas e visualizar a reconstrução da árvore filogenética, que aponta para outros genes *cry* que tenham ancestrais comuns com a sequência alvo, utilizando um serviço disponível do software MUSCLE;
4. O usuário também pode escolher qualquer uma das sequências listadas e encontrar o alinhamento dessa sequência com plasmídeos que relatam sequências de genes *cry* com alta identidade, utilizando um serviço disponível do software BLAST.

Os resultados experimentais, apresentados no Capítulo 5 e Apêndice A, indicam que o perfil HMM e as ferramentas de bioinformática combinadas cumprem a proposta de apontar sequências candidatas a serem genes *cry*. A ferramenta está disponível em um ambiente *web* e apresenta resultados legíveis ao usuário.

Vale ressaltar que no caso dos genes *cry* nós temos vários complicadores que não permitem apontar uma decisão definitiva e de imediato utilizando apenas um algoritmo, seja ele qual for, porque:

- Temos muitas sequências diferentes.
- Estas sequências diferentes conferem características diferentes.

- Não existe um padrão bem evidenciado das diferenças que justifique estas características, ou seja, não há um motivo relacionado.
- O gene *cry* não está sempre no mesmo contexto genético. E possível encontrá-lo nos plasmídeos, no cromossomo, ou nos dois locais.
- Por último, novos genes *cry* continuam a serem encontrados, o que implica na atualização constante da base de dados de modelos.

No estágio atual a ferramenta disponibiliza os resultados mais bem classificados na busca de sequências de nucleotídeos utilizando o perfil HMM. A saída para o usuário leva em consideração as informações inicial e final da localização de cada sequência alvo. A ferramenta permite a integração com os software BLAST e MUSCLE, responsáveis pelos alinhamentos da sequência e a reconstrução da árvore filogenética, respectivamente.

Na revisão e evolução da ferramenta é possível permitir ao usuário calibrar o número *bias*, para ajuste tendencioso aplicado ao *bit score*. Outra possibilidade é permitir que o usuário limite o tamanho das sequências em função das coordenadas de início e fim para cada alinhamento. O *pipeline* poderá exibir ao usuário e salvar em arquivo a sequência de nucleotídeos do provável gene *cry*, levando à proteína Cry resultante. Também é possível solicitar ao usuário um endereço de e-mail, ao invés de utilizar um endereço padrão, para que o mesmo seja informado em caso de indisponibilidade ou haja algum problema com o serviço que afete o *job* submetido.

Outros parâmetros disponíveis para o MUSCLE, como formato, árvore, ordem e sequência, como apresentados na Tabela 9 também poderão ser implementados na revisão da ferramenta.

Tabela 9 – Detalhes dos formatos aceitos pelo MUSCLE

Rótulo	Valor	Padrão	Descrição
Pearson/FASTA	fasta	false	A ferramenta permite escolher esse formato
ClustalW	clwstrict	false	Formato de alinhamento ClustalW sem numeração de base/resíduo
HTML	html	false	Alinhamento colorido no formato HTML
GCG MSF	msf	false	Formato de alinhamento de arquivo de sequência múltipla GCG
Phylip entrelaçado	phyi	false	Formato de alinhamento entrelaçado PHYLIP

Os resultados dos testes mostraram que a ferramenta cumpriu a proposta de classificar e categorizar um conjunto de sequências, além de conseguir trabalhar com grandes volumes de dados de sequências, representando um genoma inteiro. Mas é importante ressaltar a necessidade de analisar os resultados utilizando um genoma quebrado, para verificar, por exemplo, se há alguma limitação nas respostas.

REFERÊNCIAS

ABREU, I. L. d. Identificação e caracterização de um gene *cry* recombinante de *Bacillus thuringiensis* var. *Londrina*. Universidade Estadual Paulista (UNESP), 2006. Citado 2 vezes nas páginas 11 e 33.

ADANG, M. J.; CRICKMORE, N. Diversity of *Bacillus thuringiensis* crystal toxins and mechanism. *Insect Midgut and Insecticidal Proteins*, Academic Press, v. 47, p. 39, 2014. Citado 17 vezes nas páginas 11, 21, 22, 23, 25, 27, 28, 30, 31, 32, 33, 34, 35, 36, 38, 41 e 42.

ALBERTS, B. et al. *Biologia Molecular da Célula*. [S.l.]: Artmed Editora, 2009. Citado na página 27.

ALGAER, Y. Unipro ugene manual version 1.21.0. Unipro UGENE, 2016. Citado na página 67.

ALTSCHUL, S. F. et al. Basic local alignment search tool. *Journal of molecular biology*, Elsevier, v. 215, n. 3, p. 403–410, 1990. Citado 3 vezes nas páginas 59, 60 e 61.

BAO, R. et al. Review of current methods, applications, and data management for the bioinformatics analysis of whole exome sequencing. *Cancer informatics*, Libertas Academica, v. 13, n. Suppl 2, p. 67, 2014. Citado 2 vezes nas páginas 45 e 69.

BRAVO, A.; GILL, S. S.; SOBERÓN, M. Mode of action of *Bacillus thuringiensis* cry and cyt toxins and their potential for insect control. *Toxicon*, Elsevier, v. 49, n. 4, p. 423–435, 2007. Citado 4 vezes nas páginas 23, 24, 33 e 35.

BRAVO, A. et al. Evolution of *Bacillus thuringiensis* cry toxins insecticidal activity. *Microbial biotechnology*, Wiley Online Library, v. 6, n. 1, p. 17–26, 2013. Citado 9 vezes nas páginas 21, 22, 23, 24, 29, 30, 31, 40 e 42.

BRAVO, A. et al. *Bacillus thuringiensis*: a story of a successful bioinsecticide. *Insect biochemistry and molecular biology*, Elsevier, v. 41, n. 7, p. 423–431, 2011. Citado 16 vezes nas páginas 21, 22, 23, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40 e 41.

BRAVO, A. et al. Characterization of *cry* genes in a mexican *Bacillus thuringiensis* strain collection. *Applied and Environmental Microbiology*, Am Soc Microbiol, v. 64, n. 12, p. 4965–4972, 1998. Citado 7 vezes nas páginas 22, 25, 28, 32, 37, 42 e 47.

BUSO, G. Marcadores moleculares e análise filogenética. *Embrapa Recursos Genéticos e Biotecnologia. Documentos*, Brasília: Embrapa Recursos Genéticos e Biotecnologia., 2005. Citado na página 57.

CARNEIRO, A. A. C. et al. *Milho Bt: teoria e prática da produção de plantas transgênicas resistentes a insetos-praga*. [S.l.]: Embrapa Milho e Sorgo, 2009. Citado na página 41.

CARRIÇO, J. Visualizaçao e algoritmos de clustering para análise filogenética. Citado 3 vezes nas páginas 11, 58 e 59.

COKELAER, T. et al. *Flask Web Development, One Drop at a Time*. 2016. [Acessado em 18/08/2017]. Disponível em: <https://pythonhosted.org/bioservices/>. Citado 2 vezes nas páginas 64 e 75.

CONSTANSKI, K. C. et al. Seleção e caracterização molecular de isolados de bacillus thuringiensis para o controle de spodoptera spp. *Pesquisa Agropecuária Brasileira*, v. 50, n. 8, p. 730–733, 2015. Citado na página 31.

CRICKMORE, N. et al. *Bacillus thuringiensis Toxin Nomenclature*. 2016. <http://www.btnomenclature.info/>. [Acessado em 04/04/2016]. Citado na página 67.

CUSTODIO, C. J. S. et al. Fatores que contribuíram para o crescimento da produtividade do milho no brasil. *Revista Eletrônica Interdisciplinar*, v. 1, n. 15, 2016. Citado na página 30.

D'ANTONIO, M. et al. Wep: a high-performance analysis pipeline for whole-exome data. *BMC bioinformatics*, BioMed Central, v. 14, n. S7, p. S11, 2013. Citado na página 45.

EDDY, S. Hmmer user's guide. *Department of Genetics, Washington University School of Medicine*, v. 2, n. 1, 1992. Citado 10 vezes nas páginas 13, 47, 48, 49, 50, 68, 70, 73, 74 e 84.

EDDY, S. R. Profile hidden markov models. *Bioinformatics (Oxford, England)*, v. 14, n. 9, p. 755–763, 1998. Citado 4 vezes nas páginas 47, 48, 49 e 88.

EDDY, S. R. Accelerated profile hmm searches. *PLoS computational biology*, Public Library of Science, v. 7, n. 10, p. e1002195, 2011. Citado 2 vezes nas páginas 47 e 59.

EDGAR, R. C. Muscle: a multiple sequence alignment method with reduced time and space complexity. *BMC bioinformatics*, BioMed Central, v. 5, n. 1, p. 1, 2004. Citado 5 vezes nas páginas 53, 54, 55, 56 e 59.

EDGAR, R. C. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, Oxford University Press, v. 32, n. 5, p. 1792–1797, 2004. Citado 5 vezes nas páginas 47, 53, 54, 55 e 56.

ENTENDENDO Filogenia. 2016. <http://www.ib.usp.br/evosite/evo101/IIBPhylogenies.shtml>. [Acessado em 04/04/2016]. Citado na página 56.

FINN, R. D.; CLEMENTS, J.; EDDY, S. R. Hmmer web werver: Interactive sequence similarity searching. *Nucleic acids research*, Oxford Univ Press, p. gkr367, 2011. Citado 3 vezes nas páginas 47, 83 e 84.

JARGAS, A. M. *Shell Script Profissional*. [S.l.]: Novatec Editora, 2008. Citado na página 72.

JOHNSON, L. S.; EDDY, S. R.; PORTUGALY, E. Hidden markov model speed heuristic and iterative hmm search procedure. *BMC bioinformatics*, BioMed Central, v. 11, n. 1, p. 431, 2010. Citado na página 49.

KARPLUS, K.; BARRETT, C.; HUGHEY, R. Hidden markov models for detecting remote protein homologies. *Bioinformatics (Oxford, England)*, v. 14, n. 10, p. 846–856, 1998. Citado na página 49.

KIMURA, M.; WEISS, G. H. The stepping stone model of population structure and the decrease of genetic correlation with distance. *Genetics*, Genetics Society of America, v. 49, n. 4, p. 561, 1964. Citado na página 57.

LEWIN, B.; LEWIN, B. *genes IX*. [S.l.: s.n.], 2008. Citado na página 43.

LI, W. et al. The embl-ebi bioinformatics web and programmatic tools framework. *Nucleic acids research*, Oxford University Press, v. 43, n. W1, p. W580–W584, 2015. Citado 2 vezes nas páginas 45 e 46.

LOPES, D. P. *Filogenia molecular de Tunicata com ênfase em Ascidiacea*. Tese (Doutorado), 2006. Citado 3 vezes nas páginas 57, 58 e 59.

MAAGD, R. A. de; BRAVO, A.; CRICKMORE, N. How *Bacillus thuringiensis* has evolved specific toxins to colonize the insect world. *TRENDS in Genetics*, Elsevier, v. 17, n. 4, p. 193–199, 2001. Citado 10 vezes nas páginas 21, 22, 27, 28, 29, 31, 32, 33, 35 e 57.

MACIEL, S. V. P. de A. *Caracterização genética de caprinos Gurguéia no Estado do Piauí*. Tese (Doutorado) — Universidade Federal do Piauí, 2011. Citado na página 57.

MADDEN, T. The blast sequence analysis tool. National Center for Biotechnology Information (US), 2013. Citado na página 61.

MAGALHÃES, M. T. Q. d. Toxinas cry: Perspectivas para a obtenção de algodão transgênico brasileiro. 2006. Citado 2 vezes nas páginas 35 e 36.

MARUCCI, E. A. Paralelização da ferramenta de alinhamento de sequências muscle para um ambiente distribuído. Universidade Estadual Paulista (UNESP), 2009. Citado 3 vezes nas páginas 11, 54 e 55.

MCWILLIAM, H. et al. Analysis tool web services from the embl-ebi. *Nucleic acids research*, Oxford Univ Press, v. 41, n. W1, p. W597–W600, 2013. Citado 2 vezes nas páginas 45 e 46.

MONNERAT, R. et al. Genetic variability of *Spodoptera frugiperda* smith (lepidoptera: Noctuidae) populations from latin america is associated with variations in susceptibility to *Bacillus thuringiensis* cry toxins. *Applied and environmental microbiology*, Am Soc Microbiol, v. 72, n. 11, p. 7029–7035, 2006. Citado 5 vezes nas páginas 21, 23, 27, 28 e 37.

MONQUERO, P. A. Plantas transgênicas resistentes aos herbicidas: Situação e perspectivas. *Bragantia*, SciELO Brasil, v. 64, n. 4, p. 517–531, 2005. Citado na página 39.

MUSCLE (REST). 2017. <http://www.ebi.ac.uk/Tools/webservices/services/msa/muscle{_}rest{#}muscle{_}rest>. [Acessado em 15/07/2017]. Citado na página 53.

NEVES, J. C. *Programação Shell Linux-7ª edição*. [S.l.]: Brasport, 2008. Citado na página 72.

NOGUERA, P. A.; IBARRA, J. E. Detection of new *cry* genes of *Bacillus thuringiensis* by use of a novel pcr primer system. *Applied and environmental microbiology*, Am Soc Microbiol, v. 76, n. 18, p. 6150–6155, 2010. Citado 10 vezes nas páginas 21, 23, 24, 25, 28, 29, 37, 38, 40 e 42.

OKONECHNIKOV, K. et al. Unipro ugene: a unified bioinformatics toolkit. *Bioinformatics*, Oxford Univ Press, v. 28, n. 8, p. 1166–1167, 2012. Citado na página 67.

OLIVEIRA, C. T. d. *Método para melhoria da eficiência na identificação computacional de RNAs não-codificantes*. Tese (Doutorado) — Universidade de São Paulo, 2009. Citado na página 48.

PALMA, L. et al. *Bacillus thuringiensis* toxins: an overview of their biocidal activity. *Toxins*, Multidisciplinary Digital Publishing Institute, v. 6, n. 12, p. 3296–3325, 2014. Citado 12 vezes nas páginas 11, 21, 22, 23, 25, 27, 28, 29, 31, 32, 42 e 47.

PARDO-LOPEZ, L.; SOBERÓN, M.; BRAVO, A. *Bacillus thuringiensis* insecticidal three-domain cry toxins: Mode of action, insect resistance and consequences for crop protection. *FEMS microbiology reviews*, The Oxford University Press, v. 37, n. 1, p. 3–22, 2013. Citado 8 vezes nas páginas 21, 22, 23, 28, 29, 30, 35 e 39.

PINTO, L. M. N. et al. Toxinas de *Bacillus thuringiensis*. *Biotecnolog. Cienc. Desenvolv*, v. 38, p. 24–31, 2003. Citado 2 vezes nas páginas 29 e 32.

QUINLAN, A. *Bedtools Tutorial*. 2017. [Acessado em 05/10/2017]. Disponível em: <http://quinlanlab.org/tutorials/bedtools/bedtools.html>. Citado 3 vezes nas páginas 51, 52 e 53.

QUINLAN, A.; KINDLON, N. *Bedtools: a powerful toolset for genome arithmetic*. 2017. [Acessado em 05/10/2017]. Disponível em: <http://quinlanlab.org/tutorials/bedtools/bedtools.html>. Citado 4 vezes nas páginas 11, 50, 51 e 52.

RICIETO, A. P. S. et al. Effect of vegetation on the presence and genetic diversity of bacillus thuringiensis in soil. *Canadian journal of microbiology*, NRC Research Press, v. 59, n. 1, p. 28–33, 2012. Citado na página 81.

RICIETTO, A. P. S. *Análise funcional de proteínas Cry e Vip de Bacillus thuringiensis*. Tese (PhD em Genética e Biologia Molecular) — Universidade Estadual de Londrina (UEL), Centro de Ciências Biológicas, Londrina, Paraná, Brasil, 2017. Citado 2 vezes nas páginas 81 e 83.

ROH, J. Y. et al. *Bacillus thuringiensis* as a specific, safe, and effective tool for insect pest control. *Journal of microbiology and biotechnology*, THE KOREAN SOCIETY FOR APPLIED MICROBIOLOGY, v. 17, n. 4, p. 547, 2007. Citado 19 vezes nas páginas 21, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 38, 39, 41 e 42.

RONACHER, A. *Flask Web Development, One Drop at a Time*. 2017. Disponível em: <http://flask.pocoo.org/>. Citado na página 64.

SAITOU, N.; NEI, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, SMBE, v. 4, n. 4, p. 406–425, 1987. Citado na página 58.

SCHNEPF, E. et al. *Bacillus thuringiensis* and its pesticidal crystal proteins. *Microbiology and molecular biology reviews*, Am Soc Microbiol, v. 62, n. 3, p. 775–806, 1998. Citado 14 vezes nas páginas 21, 22, 23, 28, 29, 32, 35, 36, 37, 38, 39, 40, 41 e 47.

SIMPSON, J. T. et al. Abyss: a parallel assembler for short read sequence data. *Genome research*, Cold Spring Harbor Lab, v. 19, n. 6, p. 1117–1123, 2009. Citado na página 54.

SNUSTAD, D. P. *Principles of genetics*. [S.l.]: John Wiley & Sons, 2015. Citado 2 vezes nas páginas 56 e 57.

TANIGUTI, L. M. *Propagação semi-automática de termos Gene Ontology a proteínas com potencial biotecnológico para a produção de bioenergia*. Tese (Doutorado) — Escola Superior de Agricultura "Luiz de Queiroz, 2014. Citado na página 47.

TATUSOVA, T. A.; MADDEN, T. L. Blast 2 sequences, a new tool for comparing protein and nucleotide sequences. *FEMS microbiology letters*, The Oxford University Press, v. 174, n. 2, p. 247–250, 1999. Citado 3 vezes nas páginas 59, 60 e 61.

VILAS-BOAS, G.; PERUCA, A.; ARANTES, O. Biology and taxonomy of *Bacillus cereus*, *Bacillus anthracis*, and *Bacillus thuringiensis*. *Canadian journal of microbiology*, NRC Research Press, v. 53, n. 6, p. 673–687, 2007. Citado 11 vezes nas páginas 21, 22, 23, 27, 28, 29, 30, 32, 37, 39 e 40.

WEB Services at EMBL-EBI. 2017. <http://www.ebi.ac.uk/Tools/webservices/tutorials/doku.php?id=about:webservices>. [Acessado em 14/07/2017]. Citado na página 45.

WISTRAND, M.; SONNHAMMER, E. L. Improved profile hmm performance by assessment of critical algorithmic features in sam and hmmer. *BMC bioinformatics*, BioMed Central, v. 6, n. 1, p. 99, 2005. Citado 2 vezes nas páginas 47 e 49.

YE, W. et al. Mining new crystal protein genes from *Bacillus thuringiensis* on the basis of mixed plasmid-enriched genome sequencing and a computational pipeline. *Applied and environmental microbiology*, Am Soc Microbiol, v. 78, n. 14, p. 4795–4801, 2012. Citado 14 vezes nas páginas 21, 22, 23, 24, 25, 27, 28, 30, 31, 37, 38, 42, 43 e 44.

ZAHA, A.; FERREIRA, H. B.; PASSAGLIA, L. M. *Biologia Molecular Básica-5*. [S.l.]: Artmed Editora, 2014. Citado na página 27.

APÊNDICE A – TABELAS DE RESULTADOS

Este apêndice tem a intenção de mostrar ao leitor as tabelas resultantes dos testes realizados para análise dos algoritmos utilizados na ferramenta. Foram realizados três testes diferentes analisando o genoma do *Bacillus thuringiensis* BR145 contra a base de dados de perfil HMM de genes *cry*.

No primeiro teste foi considerado a totalidade da base de dados. Já no segundo teste foram retiradas 20% da totalidade da base de dados, no entanto, foram mantidos todos os grupos. No último teste também foi retirada a mesma quantidade de sequências, só que essas representavam grupos inteiros que estavam presentes nos resultados preliminares.

O genoma do *Bacillus thuringiensis* BR145 consiste em 235 *contigs*, com um tamanho total de 6.350.733 bp e 3.042.174 *reads*.

A.1 Primeiro Teste

Foram analisados os *contigs* do BR145 contra a base de dados curada de genes *cry* original, com 699 sequências. A Figura 27 demonstra a saída parcial produzida pelo *nhmmscan* e a Figura 28 se refere a uma das saídas produzidas pelo *script filter-recode*.

A.2 Segundo Teste

Apresenta a análise dos mesmos *contigs* do BR145 contra parte da base de dados curada de genes *cry* parcial, contendo 559 sequências e mantendo todos os grupos. A Figura 29 apresenta a saída parcial produzida pelo *nhmmscan* e a Figura 30 apresenta uma das saídas produzidas pelo *script filter-recode*.

A.3 Terceiro Teste

Analisa os resultados obtidos dos *contigs* do BR145 contra parte da base de dados curada de genes *cry* parcial, com 559 sequências. Porém, nesse teste foram retiradas sequências que conhecidas presentes no resultado original e que representavam grupos inteiros de família de genes *cry*. A Figura 29 apresenta a saída parcial produzida pela execução do *nhmmscan* e a Figura 30 apresenta uma das saídas produzidas pelo *script filter-recode*.

# target name	accession	query name	accession	hmmfrom	hmm to	alifrom	ali to	envfrom	env to	modlen	strand	E-value	score	bias	description of target
Cry34Aa	-	NODE_4_length_185155_cov_190.951	-	108	163	141452	141519	141431	141540	360	+	3.7	11.5	3.3	-
Cry35Ba	-	NODE_7_length_146872_cov_132.479	-	147	285	9162	9290	9139	9311	1164	+	0.62	11.2	9.1	-
Cry34Ba	-	NODE_7_length_146872_cov_132.479	-	203	310	51506	51390	51535	51368	399	-	1.5	12.6	7.9	-
Cry30Aa	-	NODE_11_length_133492_cov_140.468	-	709	925	109417	109214	109447	109189	3504	-	0.27	9.3	23.0	-
Cry30Aa	-	NODE_18_length_108561_cov_132.893	-	881	979	17503	17405	17525	17385	3504	-	4.6	4.9	18.1	-
Cry30Aa	-	NODE_22_length_93611_cov_171.592	-	1043	1203	24508	24624	24486	24662	3504	+	4.5	4.8	16.4	-
Cry34Ac	-	NODE_25_length_82350_cov_196.987	-	129	178	35100	35045	35121	35023	372	-	7.9	9.3	0.7	-
Cry29Aa	-	NODE_37_length_48629_cov_194.03	-	1143	1245	16674	16776	16653	16798	1953	+	1.7	7.1	5.1	-
Cry1H-like	-	NODE_40_length_43792_cov_181.3	-	141	252	15532	15419	15553	15399	421	-	3.5	9.2	6.9	-
Cry73Aa	-	NODE_59_length_27247_cov_379.006	-	1916	2394	22847	22378	22933	22358	2409	-	2.3e-41	140.7	49.9	-
Cry42Aa	-	NODE_59_length_27247_cov_379.006	-	1995	2424	22807	22375	22850	22367	2433	-	5.8e-33	112.8	49.0	-
Cry41Ba2	-	NODE_59_length_27247_cov_379.006	-	2070	2513	22814	22373	22840	22366	2520	-	2.1e-25	87.7	50.8	-
Cry41Aa	-	NODE_59_length_27247_cov_379.006	-	2030	2472	22799	22378	22819	22372	2478	-	4e-21	73.9	50.9	-
Cry41Ab	-	NODE_59_length_27247_cov_379.006	-	2050	2484	22797	22378	22817	22372	2490	-	1e-18	65.8	51.7	-
Cry32Wa	-	NODE_59_length_27247_cov_379.006	-	1945	2287	22823	22485	22852	22462	2400	-	8e-09	33.0	33.9	-
Cry46Aa	-	NODE_61_length_25976_cov_143.362	-	195	352	3200	3356	3181	3378	1017	+	9.2e-06	24.8	10.5	-
Cry46Ab	-	NODE_61_length_25976_cov_143.362	-	87	281	3164	3357	3144	3380	915	+	5.3e-05	22.9	17.8	-
Cry60Aa	-	NODE_67_length_23434_cov_215.945	-	579	644	23325	23260	23348	23240	912	-	0.13	11.5	2.4	-
Cry64Ba	-	NODE_71_length_22379_cov_167.84	-	33	104	13063	13134	13041	13154	870	+	6.4	5.8	2.7	-
Cry34Ab	-	NODE_72_length_22303_cov_155.942	-	150	227	21264	21333	21243	21359	372	+	4	8.7	3.8	-
Cry2Ab	-	NODE_84_length_19137_cov_360.453	-	1	1901	10288	12188	10288	12189	1902	+	0	2172.2	131.0	-
Cry2Ah	-	NODE_84_length_19137_cov_360.453	-	1	1898	10288	12188	10288	12189	1899	+	0	1979.6	131.1	-
Cry2Ad	-	NODE_84_length_19137_cov_360.453	-	1	1901	10288	12188	10288	12189	1902	+	0	1911.7	131.1	-
Cry2Ag	-	NODE_84_length_19137_cov_360.453	-	2	1904	10289	12188	10288	12189	1905	+	0	1843.9	130.9	-
Cry2Aa	-	NODE_84_length_19137_cov_360.453	-	1	1901	10288	12188	10288	12189	1902	+	0	1782.6	131.0	-
Cry2Af	-	NODE_84_length_19137_cov_360.453	-	1	1880	10288	12188	10288	12189	1881	+	0	1634.9	132.1	-
Cry2Ak	-	NODE_84_length_19137_cov_360.453	-	2	1904	10289	12188	10288	12189	1905	+	0	1570.2	130.7	-
Cry2Ac	-	NODE_84_length_19137_cov_360.453	-	1	1871	10288	12188	10288	12189	1872	+	0	1498.7	133.2	-
Cry2Al	-	NODE_84_length_19137_cov_360.453	-	2	1901	10289	12188	10288	12189	1902	+	0	1488.9	130.8	-
Cry2Ai	-	NODE_84_length_19137_cov_360.453	-	2	1900	10289	12187	10288	12189	1902	+	0	1480.2	130.7	-
Cry2Ba	-	NODE_84_length_19137_cov_360.453	-	1	1874	10288	12188	10288	12189	1875	+	0	1249.9	133.3	-
Cry18Aa	-	NODE_84_length_19137_cov_360.453	-	288	2118	10365	12186	10316	12189	2121	+	2.3e-59	200.2	118.7	-
Cry18Ba	-	NODE_84_length_19137_cov_360.453	-	131	2026	10329	12187	10307	12189	2028	+	3.6e-56	189.4	119.5	-
Cry18Ca	-	NODE_84_length_19137_cov_360.453	-	187	1951	10336	12055	10314	12077	2088	+	3.9e-52	176.3	107.1	-
Cry1Cb	-	NODE_84_length_19137_cov_360.453	-	3395	3530	7715	7850	7695	7851	3531	+	1e-13	47.4	15.3	-
Cry1Fb	-	NODE_84_length_19137_cov_360.453	-	3369	3506	7715	7850	7694	7851	3507	+	4.2e-12	41.8	15.9	-
Cry1Ja	-	NODE_84_length_19137_cov_360.453	-	3367	3503	7714	7850	7691	7851	3504	+	9.1e-12	40.9	15.7	-
Cry1Jb	-	NODE_84_length_19137_cov_360.453	-	3380	3512	7718	7850	7697	7851	3513	+	1.2e-11	40.7	15.4	-
Cry1Ba	-	NODE_84_length_19137_cov_360.453	-	3493	3623	7720	7850	7700	7851	3624	+	2e-11	39.8	15.1	-
Cry1Be	-	NODE_84_length_19137_cov_360.453	-	3474	3605	7719	7850	7698	7851	3606	+	3.1e-11	39.2	15.2	-
Cry1Da	-	NODE_84_length_19137_cov_360.453	-	3360	3497	7713	7850	7692	7851	3498	+	4.7e-11	38.5	15.8	-
Cry1Db	-	NODE_84_length_19137_cov_360.453	-	3344	3483	7714	7851	7694	7851	3483	+	6.1e-11	38.1	16.2	-
Cry1Bb	-	NODE_84_length_19137_cov_360.453	-	3481	3611	7720	7850	7700	7851	3612	+	6.6e-11	38.1	14.9	-
Cry1Jd	-	NODE_84_length_19137_cov_360.453	-	3369	3506	7713	7850	7692	7851	3507	+	8.5e-11	37.9	16.6	-
Cry1Ac	-	NODE_84_length_19137_cov_360.453	-	3393	3534	7710	7851	7689	7851	3534	+	9e-11	37.4	16.7	-
Cry9Ga	-	NODE_84_length_19137_cov_360.453	-	3549	3688	7711	7849	7690	7851	3690	+	9.1e-11	37.7	16.3	-
Cry1Aa	-	NODE_84_length_19137_cov_360.453	-	3392	3530	7712	7850	7690	7851	3531	+	1.3e-10	36.8	16.3	-
Cry1Ka	-	NODE_84_length_19137_cov_360.453	-	3427	3569	7708	7850	7688	7851	3570	+	2.1e-10	36.3	16.0	-
Cry1Gb	-	NODE_84_length_19137_cov_360.453	-	3373	3510	7714	7851	7694	7851	3510	+	2.5e-10	36.0	15.3	-
Cry1Ai	-	NODE_84_length_19137_cov_360.453	-	3406	3545	7711	7850	7689	7851	3546	+	3.6e-10	35.3	15.8	-
Cry1Fa	-	NODE_84_length_19137_cov_360.453	-	3388	3524	7714	7850	7694	7851	3525	+	4.4e-10	35.4	15.1	-
Cry1Bf	-	NODE_84_length_19137_cov_360.453	-	3554	3686	7718	7850	7698	7851	3687	+	4.5e-10	35.5	15.3	-
Cry1Bi	-	NODE_84_length_19137_cov_360.453	-	3581	3713	7718	7850	7698	7851	3714	+	6.5e-10	34.9	15.3	-
Cry1Bg	-	NODE_84_length_19137_cov_360.453	-	3575	3707	7718	7850	7698	7851	3708	+	1.2e-09	34.1	15.2	-
Cry1Ab	-	NODE_84_length_19137_cov_360.453	-	3331	3467	7714	7850	7693	7851	3468	+	1.6e-09	33.1	15.4	-
Cry1Ea	-	NODE_84_length_19137_cov_360.453	-	3379	3514	7714	7849	7694	7851	3516	+	2e-09	33.3	15.1	-
Cry1Ad	-	NODE_84_length_19137_cov_360.453	-	3403	3539	7714	7850	7694	7851	3540	+	2.9e-09	32.6	15.1	-
Cry1Ae	-	NODE_84_length_19137_cov_360.453	-	3409	3545	7714	7850	7693	7851	3546	+	4.5e-09	32.1	15.6	-

Figura 27 – Saída da execução do *nhmmscan* utilizando a base de dados completa referente a seção A.1.

1	NODE_110_length_12556_cov_314.781	CRY_DETECT	gene	10454	11260	3.9e-101	+	.	Name= Cry30Aa	Bit-Score: 337.0	Bias: 46.8
2	NODE_118_length_10549_cov_579.939	CRY_DETECT	gene	287	2187	0	+	.	Name= Cry2Aa	Bit-Score: 2180.3	Bias: 121.6
3	NODE_118_length_10549_cov_579.939	CRY_DETECT	gene	7361	9577	0	-	.	Name= Cry1Ia	Bit-Score: 2496.6	Bias: 115.5
4	NODE_118_length_10549_cov_579.939	CRY_DETECT	gene	10062	10549	3.2e-161	-	.	Name= Cry1Ac	Bit-Score: 535.7	Bias: 18.1
5	NODE_11_length_133492_cov_140.468	CRY_DETECT	gene	109214	109417	0.27	-	.	Name= Cry30Aa	Bit-Score: 9.3	Bias: 23.0
6	NODE_124_length_8888_cov_128.481	CRY_DETECT	gene	5117	5304	0.013	-	.	Name= Cry30Aa	Bit-Score: 9.8	Bias: 18.6
7	NODE_166_length_3504_cov_159.262	CRY_DETECT	gene	2806	2904	0.27	-	.	Name= Cry30Aa	Bit-Score: 4.1	Bias: 17.7
8	NODE_172_length_3262_cov_120.933	CRY_DETECT	gene	2995	3168	0.26	-	.	Name= Cry7Gd	Bit-Score: 4.1	Bias: 14.6
9	NODE_186_length_2444_cov_1318.21	CRY_DETECT	gene	1	2444	0	+	.	Name= Cry1Ac	Bit-Score: 2864.7	Bias: 92.0
10	NODE_18_length_108561_cov_132.893	CRY_DETECT	gene	17405	17503	4.6	-	.	Name= Cry30Aa	Bit-Score: 4.9	Bias: 18.1
11	NODE_200_length_1638_cov_2092.64	CRY_DETECT	gene	1015	1638	9e-210	+	.	Name= Cry1Aa	Bit-Score: 694.0	Bias: 31.6
12	NODE_22_length_93611_cov_171.592	CRY_DETECT	gene	24508	24624	4.5	+	.	Name= Cry30Aa	Bit-Score: 4.8	Bias: 16.4
13	NODE_234_length_1013_cov_766.226	CRY_DETECT	gene	1	1013	0	-	.	Name= Cry1Ab	Bit-Score: 1153.5	Bias: 42.4
14	NODE_235_length_1010_cov_839.119	CRY_DETECT	gene	1	1010	0	+	.	Name= Cry1Aa	Bit-Score: 1175.7	Bias: 33.3
15	NODE_250_length_872_cov_1616.63	CRY_DETECT	gene	1	872	2.5e-301	+	.	Name= Cry1Aa	Bit-Score: 996.8	Bias: 44.8
16	NODE_256_length_807_cov_903.879	CRY_DETECT	gene	1	506	2.7e-163	+	.	Name= Cry1Ab	Bit-Score: 538.7	Bias: 17.5
17	NODE_25_length_82350_cov_196.987	CRY_DETECT	gene	35045	35100	7.9	-	.	Name= Cry34Ac	Bit-Score: 9.3	Bias: 0.7
18	NODE_262_length_735_cov_883.235	CRY_DETECT	gene	1	735	1.8e-255	-	.	Name= Cry1Ab	Bit-Score: 844.3	Bias: 26.7
19	NODE_350_length_471_cov_1.02326	CRY_DETECT	gene	108	259	0.026	-	.	Name= Cry2Af	Bit-Score: 6.0	Bias: 9.5
20	NODE_358_length_466_cov_432.563	CRY_DETECT	gene	22	231	0.023	+	.	Name= Cry20Aa	Bit-Score: 5.9	Bias: 19.1
21	NODE_358_length_466_cov_432.563	CRY_DETECT	gene	309	449	0.17	+	.	Name= Cry20Aa	Bit-Score: 3.0	Bias: 13.8
22	NODE_37_length_48629_cov_194.03	CRY_DETECT	gene	16674	16776	1.7	+	.	Name= Cry29Aa	Bit-Score: 7.1	Bias: 5.1
23	NODE_40_length_43792_cov_181.3	CRY_DETECT	gene	15419	15532	3.5	-	.	Name= Cry1H-like	Bit-Score: 9.2	Bias: 6.9
24	NODE_4_length_185155_cov_190.951	CRY_DETECT	gene	141452	141519	3.7	+	.	Name= Cry34Aa	Bit-Score: 11.5	Bias: 3.3
25	NODE_59_length_27247_cov_379.006	CRY_DETECT	gene	22373	22847	2.3e-41	-	.	Name= Cry73Aa	Bit-Score: 140.7	Bias: 49.9
26	NODE_606_length_380_cov_4156.68	CRY_DETECT	gene	1	380	4.5e-130	-	.	Name= Cry1Ac	Bit-Score: 427.5	Bias: 8.1
27	NODE_61_length_25976_cov_143.362	CRY_DETECT	gene	3164	3357	9.2e-06	+	.	Name= Cry46Aa	Bit-Score: 24.8	Bias: 10.5
28	NODE_640_length_245_cov_986.542	CRY_DETECT	gene	1	245	2.2e-79	-	.	Name= Cry1Aa	Bit-Score: 258.7	Bias: 10.8
29	NODE_655_length_158_cov_439.774	CRY_DETECT	gene	17	143	0.25	-	.	Name= Cry20Aa	Bit-Score: 0.9	Bias: 10.2
30	NODE_657_length_154_cov_44.0741	CRY_DETECT	gene	20	140	1.2	-	.	Name= Cry20Aa	Bit-Score: -1.4	Bias: 10.9
31	NODE_659_length_133_cov_839.833	CRY_DETECT	gene	24	129	6.7	-	.	Name= Cry20Aa	Bit-Score: -4.0	Bias: 9.6
32	NODE_67_length_23434_cov_215.945	CRY_DETECT	gene	23260	23325	0.13	-	.	Name= Cry60Aa	Bit-Score: 11.5	Bias: 2.4
33	NODE_71_length_22379_cov_167.84	CRY_DETECT	gene	13063	13134	6.4	+	.	Name= Cry64Ba	Bit-Score: 5.8	Bias: 2.7
34	NODE_72_length_22303_cov_155.942	CRY_DETECT	gene	21264	21333	4	+	.	Name= Cry34Ab	Bit-Score: 8.7	Bias: 3.8
35	NODE_7_length_146872_cov_132.479	CRY_DETECT	gene	9162	9290	0.62	+	.	Name= Cry35Ba	Bit-Score: 11.2	Bias: 9.1
36	NODE_7_length_146872_cov_132.479	CRY_DETECT	gene	51390	51506	1.5	-	.	Name= Cry34Ba	Bit-Score: 12.6	Bias: 7.9
37	NODE_84_length_19137_cov_360.453	CRY_DETECT	gene	7708	7853	1e-13	+	.	Name= Cry1Cb	Bit-Score: 47.4	Bias: 15.3
38	NODE_84_length_19137_cov_360.453	CRY_DETECT	gene	10288	12188	0	+	.	Name= Cry2Ab	Bit-Score: 2172.2	Bias: 131.0
39	NODE_87_length_18294_cov_409.757	CRY_DETECT	gene	1434	1632	1.4e-28	+	.	Name= Cry30Aa	Bit-Score: 96.9	Bias: 12.1
40	NODE_87_length_18294_cov_409.757	CRY_DETECT	gene	15178	15719	2.4e-47	+	.	Name= Cry30Aa	Bit-Score: 159.1	Bias: 30.9
41	NODE_94_length_15414_cov_99.2846	CRY_DETECT	gene	6355	6662	7.7e-05	+	.	Name= Cry22Bb	Bit-Score: 21.3	Bias: 16.4
42	NODE_95_length_15200_cov_205.991	CRY_DETECT	gene	6134	6244	0.11	+	.	Name= Cry5Ad	Bit-Score: 8.0	Bias: 7.4
43	NODE_96_length_15085_cov_216.286	CRY_DETECT	gene	12514	12611	2.1	-	.	Name= Cry34Aa	Bit-Score: 8.7	Bias: 4.5

Figura 28 – Informações do genes *cry* encontrados referente a seção A.1.

# target name	accession	query name	accession	hmmfrom	hmm to	alifrom	ali to	envfrom	env to	modlen	strand	E-value	score	bias	description of target
Cry34Aa	-	NODE_4_length_185155_cov_190.951	-	108	163	141452	141519	141435	141539	360	+	2.8	11.7	3.1	-
Cry35Ba	-	NODE_7_length_146872_cov_132.479	-	147	285	9162	9290	9139	9311	1164	+	0.52	11.2	9.1	-
Cry34Ba	-	NODE_7_length_146872_cov_132.479	-	203	310	51506	51390	51538	51368	399	-	1.2	12.7	8.3	-
Cry1Bj1	-	NODE_9_length_135708_cov_151.664	-	1850	2051	104081	103870	104103	103848	3702	-	5.9	4.5	10.8	-
Cry30Aa	-	NODE_11_length_133492_cov_140.468	-	709	925	109417	109214	109447	109189	3504	-	0.24	9.3	23.0	-
Cry30Aa	-	NODE_18_length_108561_cov_132.893	-	881	979	17503	17405	17525	17385	3504	-	4	4.9	18.1	-
Cry30Aa	-	NODE_22_length_93611_cov_171.592	-	1043	1203	24508	24624	24486	24662	3504	+	3.9	4.8	16.4	-
Cry34Ac	-	NODE_25_length_82350_cov_196.987	-	129	177	35100	35046	35121	35025	372	-	9.2	8.8	0.8	-
Cry29Aa	-	NODE_37_length_48629_cov_194.03	-	1143	1245	16674	16776	16653	16798	1953	+	1.5	7.1	5.1	-
Cry73Aa	-	NODE_59_length_27247_cov_379.006	-	1916	2394	22847	22378	22933	22358	2409	-	2e-41	140.7	49.9	-
Cry42Aa	-	NODE_59_length_27247_cov_379.006	-	1995	2424	22807	22375	22850	22367	2433	-	5e-33	112.8	49.0	-
Cry41Ba2	-	NODE_59_length_27247_cov_379.006	-	2070	2513	22814	22373	22840	22366	2520	-	1.8e-25	87.7	50.8	-
Cry41Aa	-	NODE_59_length_27247_cov_379.006	-	2030	2472	22799	22378	22819	22372	2478	-	3.5e-21	73.9	50.9	-
Cry41Ab	-	NODE_59_length_27247_cov_379.006	-	2050	2484	22797	22378	22817	22372	2490	-	8.9e-19	65.8	51.7	-
Cry32Wa	-	NODE_59_length_27247_cov_379.006	-	1945	2287	22823	22485	22852	22462	2400	-	7e-09	33.0	33.9	-
Cry46Aa	-	NODE_61_length_25976_cov_143.362	-	195	352	3200	3356	3181	3378	1017	+	7.9e-06	24.8	10.5	-
Cry46Ab	-	NODE_61_length_25976_cov_143.362	-	87	281	3164	3357	3144	3380	915	+	4.6e-05	22.9	17.8	-
Cry60Aa	-	NODE_67_length_23434_cov_215.945	-	579	644	23325	23260	23348	23240	912	-	0.11	11.5	2.4	-
Cry34Ba	-	NODE_67_length_23434_cov_215.945	-	118	181	6760	6816	6747	6838	399	+	8.7	7.3	1.1	-
Cry64Ba	-	NODE_71_length_22379_cov_167.84	-	33	104	13063	13134	13041	13154	870	+	5.5	5.8	2.7	-
Cry34Ab	-	NODE_72_length_22303_cov_155.942	-	150	227	21264	21333	21243	21359	372	+	3.4	8.7	3.8	-
Cry2Ab	-	NODE_84_length_19137_cov_360.453	-	1	1901	10288	12188	10288	12189	1902	+	0	2170.2	131.0	-
Cry2Ah	-	NODE_84_length_19137_cov_360.453	-	1	1898	10288	12188	10288	12189	1899	+	0	1990.3	131.1	-
Cry2Ad	-	NODE_84_length_19137_cov_360.453	-	1	1901	10288	12188	10288	12189	1902	+	0	1916.5	131.1	-
Cry2Ag	-	NODE_84_length_19137_cov_360.453	-	2	1904	10289	12188	10288	12189	1905	+	0	1843.9	130.9	-
Cry2Aa	-	NODE_84_length_19137_cov_360.453	-	1	1901	10288	12188	10288	12189	1902	+	0	1726.0	131.0	-
Cry2Af	-	NODE_84_length_19137_cov_360.453	-	1	1880	10288	12188	10288	12189	1881	+	0	1634.9	132.1	-
Cry2Ak	-	NODE_84_length_19137_cov_360.453	-	2	1904	10289	12188	10288	12189	1905	+	0	1570.2	130.7	-
Cry2Ac	-	NODE_84_length_19137_cov_360.453	-	1	1871	10288	12188	10288	12189	1872	+	0	1497.0	133.2	-
Cry2Al	-	NODE_84_length_19137_cov_360.453	-	2	1901	10289	12188	10288	12189	1902	+	0	1488.9	130.8	-
Cry2Ai	-	NODE_84_length_19137_cov_360.453	-	2	1900	10289	12187	10288	12189	1902	+	0	1480.2	130.7	-
Cry2Ba	-	NODE_84_length_19137_cov_360.453	-	1	1874	10288	12188	10288	12189	1875	+	0	1249.9	133.3	-
Cry18Aa	-	NODE_84_length_19137_cov_360.453	-	288	2118	10365	12186	10316	12189	2121	+	2e-59	200.2	118.7	-
Cry18Ba	-	NODE_84_length_19137_cov_360.453	-	131	2026	10329	12187	10307	12189	2028	+	3.1e-56	189.4	119.5	-
Cry18Ca	-	NODE_84_length_19137_cov_360.453	-	187	1951	10336	12055	10314	12077	2088	+	3.3e-52	176.3	107.1	-
Cry1Cb	-	NODE_84_length_19137_cov_360.453	-	3395	3530	7715	7850	7695	7851	3531	+	8.8e-14	47.4	15.3	-
Cry1Ja	-	NODE_84_length_19137_cov_360.453	-	3367	3503	7714	7850	7691	7851	3504	+	7.9e-12	40.9	15.7	-
Cry1Fb	-	NODE_84_length_19137_cov_360.453	-	3368	3507	7714	7851	7694	7851	3507	+	7.9e-12	40.7	16.0	-
Cry1Jb	-	NODE_84_length_19137_cov_360.453	-	3380	3512	7718	7850	7697	7851	3513	+	1.1e-11	40.7	15.4	-
Cry1Be	-	NODE_84_length_19137_cov_360.453	-	3474	3605	7719	7850	7698	7851	3606	+	1.5e-11	40.0	15.2	-
Cry1Ba	-	NODE_84_length_19137_cov_360.453	-	3493	3623	7720	7850	7700	7851	3624	+	2.2e-11	39.4	15.0	-
Cry1Ac	-	NODE_84_length_19137_cov_360.453	-	3396	3537	7710	7851	7689	7851	3537	+	4.5e-11	38.1	16.7	-
Cry1Da	-	NODE_84_length_19137_cov_360.453	-	3353	3490	7709	7846	7689	7848	3492	+	4.7e-11	38.3	15.6	-
Cry1Db	-	NODE_84_length_19137_cov_360.453	-	3344	3483	7714	7851	7694	7851	3483	+	5.3e-11	38.1	16.2	-
Cry1Bb	-	NODE_84_length_19137_cov_360.453	-	3481	3611	7720	7850	7700	7851	3612	+	5.7e-11	38.1	14.9	-
Cry1Aa	-	NODE_84_length_19137_cov_360.453	-	3390	3531	7710	7851	7689	7851	3531	+	6.2e-11	37.7	16.7	-
Cry1Jd	-	NODE_84_length_19137_cov_360.453	-	3369	3506	7713	7850	7692	7851	3507	+	7.3e-11	37.9	16.6	-
Cry9Ga	-	NODE_84_length_19137_cov_360.453	-	3549	3688	7711	7849	7690	7851	3690	+	7.9e-11	37.7	16.3	-
Cry1Ka	-	NODE_84_length_19137_cov_360.453	-	3427	3569	7708	7850	7688	7851	3570	+	1.8e-10	36.3	16.0	-
Cry1Gb	-	NODE_84_length_19137_cov_360.453	-	3373	3510	7714	7851	7694	7851	3510	+	2.1e-10	36.0	15.3	-
Cry1Ai	-	NODE_84_length_19137_cov_360.453	-	3406	3545	7711	7850	7689	7851	3546	+	3.1e-10	35.3	15.8	-
Cry1Bf	-	NODE_84_length_19137_cov_360.453	-	3554	3686	7718	7850	7698	7851	3687	+	3.9e-10	35.5	15.3	-
Cry1Bi	-	NODE_84_length_19137_cov_360.453	-	3581	3713	7718	7850	7698	7851	3714	+	5.6e-10	34.9	15.3	-
Cry1Bg	-	NODE_84_length_19137_cov_360.453	-	3575	3707	7718	7850	7698	7851	3708	+	1e-09	34.1	15.2	-
Cry1Fa	-	NODE_84_length_19137_cov_360.453	-	3388	3520	7714	7846	7694	7848	3522	+	1.2e-09	33.7	14.3	-
Cry1Bj1	-	NODE_84_length_19137_cov_360.453	-	3569	3701	7718	7850	7698	7851	3702	+	1.3e-09	33.8	15.3	-
Cry1Ea	-	NODE_84_length_19137_cov_360.453	-	3379	3514	7714	7849	7694	7851	3516	+	1.6e-09	33.4	15.1	-
Cry1Ad	-	NODE_84_length_19137_cov_360.453	-	3403	3539	7714	7850	7694	7851	3540	+	2.5e-09	32.6	15.1	-

Figura 29 – Saída da execução do *nhmmscan* utilizando 80% da base de dados referente a seção A.2.

1	NODE_110_length_12556_cov_314.781	CRY_DETECT	gene	10454	11260	3.4e-101	+	.	Name= Cry30Aa	Bit-Score:	337.0	Bias: 46.8
2	NODE_118_length_10549_cov_579.939	CRY_DETECT	gene	287	2187	0	+	.	Name= Cry2Aa	Bit-Score:	2182.7	Bias: 121.6
3	NODE_118_length_10549_cov_579.939	CRY_DETECT	gene	7393	9577	0	-	.	Name= Cry1Ia	Bit-Score:	2504.2	Bias: 115.5
4	NODE_118_length_10549_cov_579.939	CRY_DETECT	gene	10062	10549	8.1e-162	-	.	Name= Cry1Ac	Bit-Score:	537.4	Bias: 18.1
5	NODE_11_length_133492_cov_140.468	CRY_DETECT	gene	109214	109417	0.24	-	.	Name= Cry30Aa	Bit-Score:	9.3	Bias: 23.0
6	NODE_124_length_8888_cov_128.481	CRY_DETECT	gene	5117	5304	0.011	-	.	Name= Cry30Aa	Bit-Score:	9.8	Bias: 18.6
7	NODE_166_length_3504_cov_159.262	CRY_DETECT	gene	2806	2904	0.23	-	.	Name= Cry30Aa	Bit-Score:	4.1	Bias: 17.7
8	NODE_172_length_3262_cov_120.933	CRY_DETECT	gene	2995	3168	0.22	-	.	Name= Cry7Gd	Bit-Score:	4.1	Bias: 14.6
9	NODE_186_length_2444_cov_1318.21	CRY_DETECT	gene	1	2444	0	+	.	Name= Cry1Ac	Bit-Score:	2863.2	Bias: 92.0
10	NODE_18_length_108561_cov_132.893	CRY_DETECT	gene	17405	17503	4	-	.	Name= Cry30Aa	Bit-Score:	4.9	Bias: 18.1
11	NODE_200_length_1638_cov_2092.64	CRY_DETECT	gene	1015	1638	5.8e-208	+	.	Name= Cry1Aa	Bit-Score:	687.8	Bias: 31.6
12	NODE_22_length_93611_cov_171.592	CRY_DETECT	gene	24508	24624	3.9	+	.	Name= Cry30Aa	Bit-Score:	4.8	Bias: 16.4
13	NODE_234_length_1013_cov_766.226	CRY_DETECT	gene	1	1013	0	-	.	Name= Cry1Ab	Bit-Score:	1142.7	Bias: 42.4
14	NODE_235_length_1010_cov_839.119	CRY_DETECT	gene	1	1010	0	+	.	Name= Cry1Aa	Bit-Score:	1163.6	Bias: 33.3
15	NODE_250_length_872_cov_1616.63	CRY_DETECT	gene	1	872	1.1e-298	+	.	Name= Cry1Ab	Bit-Score:	987.9	Bias: 44.8
16	NODE_256_length_807_cov_903.879	CRY_DETECT	gene	1	506	1.3e-165	+	.	Name= Cry1Ab	Bit-Score:	546.3	Bias: 17.6
17	NODE_25_length_82350_cov_196.987	CRY_DETECT	gene	35046	35100	9.2	-	.	Name= Cry34Ac	Bit-Score:	8.8	Bias: 0.8
18	NODE_262_length_735_cov_883.235	CRY_DETECT	gene	1	735	5.9e-258	-	.	Name= Cry1Ab	Bit-Score:	852.5	Bias: 26.7
19	NODE_350_length_471_cov_1.02326	CRY_DETECT	gene	108	259	0.023	-	.	Name= Cry2Af	Bit-Score:	6.0	Bias: 9.5
20	NODE_358_length_466_cov_432.563	CRY_DETECT	gene	22	231	0.02	+	.	Name= Cry20Aa	Bit-Score:	5.9	Bias: 19.1
21	NODE_358_length_466_cov_432.563	CRY_DETECT	gene	309	449	0.15	+	.	Name= Cry20Aa	Bit-Score:	3.0	Bias: 13.8
22	NODE_37_length_48629_cov_194.03	CRY_DETECT	gene	16674	16776	1.5	+	.	Name= Cry29Aa	Bit-Score:	7.1	Bias: 5.1
23	NODE_4_length_185155_cov_190.951	CRY_DETECT	gene	141452	141519	2.8	+	.	Name= Cry34Aa	Bit-Score:	11.7	Bias: 3.1
24	NODE_59_length_27247_cov_379.006	CRY_DETECT	gene	22373	22847	2e-41	-	.	Name= Cry73Aa	Bit-Score:	140.7	Bias: 49.9
25	NODE_606_length_380_cov_4156.68	CRY_DETECT	gene	1	380	4.5e-131	-	.	Name= Cry1Ab	Bit-Score:	430.6	Bias: 8.1
26	NODE_61_length_25976_cov_143.362	CRY_DETECT	gene	3164	3357	7.9e-06	+	.	Name= Cry46Aa	Bit-Score:	24.8	Bias: 10.5
27	NODE_640_length_245_cov_986.542	CRY_DETECT	gene	1	245	1.6e-78	-	.	Name= Cry1Aa	Bit-Score:	255.6	Bias: 10.8
28	NODE_655_length_158_cov_439.774	CRY_DETECT	gene	17	143	0.22	-	.	Name= Cry20Aa	Bit-Score:	0.9	Bias: 10.2
29	NODE_657_length_154_cov_44.0741	CRY_DETECT	gene	20	140	1.1	-	.	Name= Cry20Aa	Bit-Score:	-1.4	Bias: 10.9
30	NODE_659_length_133_cov_839.833	CRY_DETECT	gene	24	129	5.8	-	.	Name= Cry20Aa	Bit-Score:	-4.0	Bias: 9.6
31	NODE_67_length_23434_cov_215.945	CRY_DETECT	gene	6760	6816	8.7	+	.	Name= Cry34Ba	Bit-Score:	7.3	Bias: 1.1
32	NODE_67_length_23434_cov_215.945	CRY_DETECT	gene	23260	23325	0.11	-	.	Name= Cry60Aa	Bit-Score:	11.5	Bias: 2.4
33	NODE_71_length_22379_cov_167.84	CRY_DETECT	gene	13063	13134	5.5	+	.	Name= Cry64Ba	Bit-Score:	5.8	Bias: 2.7
34	NODE_72_length_22303_cov_155.942	CRY_DETECT	gene	21264	21333	3.4	+	.	Name= Cry34Ab	Bit-Score:	8.7	Bias: 3.8
35	NODE_7_length_146872_cov_132.479	CRY_DETECT	gene	9162	9290	0.52	+	.	Name= Cry35Ba	Bit-Score:	11.2	Bias: 9.1
36	NODE_7_length_146872_cov_132.479	CRY_DETECT	gene	51390	51506	1.2	-	.	Name= Cry34Ba	Bit-Score:	12.7	Bias: 8.3
37	NODE_84_length_19137_cov_360.453	CRY_DETECT	gene	7708	7853	8.8e-14	+	.	Name= Cry1Cb	Bit-Score:	47.4	Bias: 15.3
38	NODE_84_length_19137_cov_360.453	CRY_DETECT	gene	10288	12188	0	+	.	Name= Cry2Ab	Bit-Score:	2170.2	Bias: 131.0
39	NODE_87_length_18294_cov_409.757	CRY_DETECT	gene	1434	1632	1.2e-28	+	.	Name= Cry30Aa	Bit-Score:	96.9	Bias: 12.1
40	NODE_87_length_18294_cov_409.757	CRY_DETECT	gene	15178	15719	2.1e-47	+	.	Name= Cry30Aa	Bit-Score:	159.1	Bias: 30.9
41	NODE_94_length_15414_cov_99.2846	CRY_DETECT	gene	6355	6662	6.6e-05	+	.	Name= Cry22Bb	Bit-Score:	21.3	Bias: 16.4
42	NODE_95_length_15200_cov_205.991	CRY_DETECT	gene	6134	6244	0.093	+	.	Name= Cry5Ad	Bit-Score:	8.0	Bias: 7.4
43	NODE_96_length_15085_cov_216.286	CRY_DETECT	gene	12514	12611	1.8	-	.	Name= Cry34Aa	Bit-Score:	8.7	Bias: 4.5
44	NODE_9_length_135708_cov_151.664	CRY_DETECT	gene	103870	104081	5.9	-	.	Name= Cry1Bj1	Bit-Score:	4.5	Bias: 10.8

Figura 30 – Informações do genes *cry* encontrados em 80% da base de dados referente a seção A.2.

# target name	accession	query name	accession	hmmfrom	hmm to	alifrom	ali to	envfrom	env to	modlen	strand	E-value	score	bias	description of target
Cry34Aa	-	NODE_4_length_185155_cov_190.951	-	108	163	141452	141519	141431	141540	360	+	3.1	11.5	3.3	-
Cry35Ba	-	NODE_7_length_146872_cov_132.479	-	147	285	9162	9290	9139	9311	1164	+	0.52	11.2	9.1	-
Cry34Ba	-	NODE_7_length_146872_cov_132.479	-	203	310	51506	51390	51535	51368	399	-	1.3	12.6	7.9	-
Cry1Bj1	-	NODE_9_length_135708_cov_151.664	-	1850	2051	104081	103870	104103	103848	3702	-	5.8	4.5	10.8	-
Cry30Aa	-	NODE_11_length_133492_cov_140.468	-	709	925	109417	109214	109447	109189	3504	-	0.23	9.3	23.0	-
Cry30Aa	-	NODE_18_length_108561_cov_132.893	-	881	979	17503	17405	17525	17385	3504	-	3.9	4.9	18.1	-
Cry30Aa	-	NODE_22_length_93611_cov_171.592	-	1043	1203	24508	24624	24486	24662	3504	+	3.8	4.8	16.4	-
Cry34Ac	-	NODE_25_length_82350_cov_196.987	-	129	178	35100	35045	35121	35023	372	-	6.6	9.3	0.7	-
Cry29Aa	-	NODE_37_length_48629_cov_194.03	-	1143	1245	16674	16776	16653	16798	1953	+	1.4	7.1	5.1	-
Cry73Aa	-	NODE_59_length_27247_cov_379.006	-	1916	2394	22847	22378	22933	22358	2409	-	1.9e-41	140.7	49.9	-
Cry42Aa	-	NODE_59_length_27247_cov_379.006	-	1995	2424	22807	22375	22850	22367	2433	-	4.9e-33	112.8	49.0	-
Cry41Ba2	-	NODE_59_length_27247_cov_379.006	-	2070	2513	22814	22373	22840	22366	2520	-	1.7e-25	87.7	50.8	-
Cry41Aa	-	NODE_59_length_27247_cov_379.006	-	2030	2472	22799	22378	22819	22372	2478	-	3.4e-21	73.9	50.9	-
Cry41Ab	-	NODE_59_length_27247_cov_379.006	-	2050	2484	22797	22378	22817	22372	2490	-	8.7e-19	65.8	51.7	-
Cry32Wa	-	NODE_59_length_27247_cov_379.006	-	1945	2287	22823	22485	22852	22462	2400	-	6.8e-09	33.0	33.9	-
Cry46Aa	-	NODE_61_length_25976_cov_143.362	-	195	352	3200	3356	3181	3378	1017	+	7.8e-06	24.8	10.5	-
Cry46Ab	-	NODE_61_length_25976_cov_143.362	-	87	281	3164	3357	3144	3380	915	+	4.5e-05	22.9	17.8	-
Cry60Aa	-	NODE_67_length_23434_cov_215.945	-	579	644	23325	23260	23348	23240	912	-	0.11	11.5	2.4	-
Cry64Ba	-	NODE_71_length_22379_cov_167.84	-	33	104	13063	13134	13041	13154	870	+	5.4	5.8	2.7	-
Cry34Ab	-	NODE_72_length_22303_cov_155.942	-	150	227	21264	21333	21243	21359	372	+	3.4	8.7	3.8	-
Cry2Ab	-	NODE_84_length_19137_cov_360.453	-	1	1901	10288	12188	10288	12189	1902	+	0	2172.2	131.0	-
Cry2Ah	-	NODE_84_length_19137_cov_360.453	-	1	1898	10288	12188	10288	12189	1899	+	0	1979.6	131.1	-
Cry2Ad	-	NODE_84_length_19137_cov_360.453	-	1	1901	10288	12188	10288	12189	1902	+	0	1911.7	131.1	-
Cry2Ag	-	NODE_84_length_19137_cov_360.453	-	2	1904	10289	12188	10288	12189	1905	+	0	1843.9	130.9	-
Cry2Aa	-	NODE_84_length_19137_cov_360.453	-	1	1901	10288	12188	10288	12189	1902	+	0	1782.6	131.0	-
Cry2Af	-	NODE_84_length_19137_cov_360.453	-	1	1880	10288	12188	10288	12189	1881	+	0	1634.9	132.1	-
Cry2Ak	-	NODE_84_length_19137_cov_360.453	-	2	1904	10289	12188	10288	12189	1905	+	0	1570.2	130.7	-
Cry2Ac	-	NODE_84_length_19137_cov_360.453	-	1	1871	10288	12188	10288	12189	1872	+	0	1498.7	133.2	-
Cry2Al	-	NODE_84_length_19137_cov_360.453	-	2	1901	10289	12188	10288	12189	1902	+	0	1488.9	130.8	-
Cry2Ai	-	NODE_84_length_19137_cov_360.453	-	2	1900	10289	12187	10288	12189	1902	+	0	1480.2	130.7	-
Cry2Ba	-	NODE_84_length_19137_cov_360.453	-	1	1874	10288	12188	10288	12189	1875	+	0	1249.9	133.3	-
Cry18Aa	-	NODE_84_length_19137_cov_360.453	-	288	2118	10365	12186	10316	12189	2121	+	2e-59	200.2	118.7	-
Cry18Ba	-	NODE_84_length_19137_cov_360.453	-	131	2026	10329	12187	10307	12189	2028	+	3.1e-56	189.4	119.5	-
Cry18Ca	-	NODE_84_length_19137_cov_360.453	-	187	1951	10336	12055	10314	12077	2088	+	3.3e-52	176.3	107.1	-
Cry1Cb	-	NODE_84_length_19137_cov_360.453	-	3395	3530	7715	7850	7695	7851	3531	+	8.6e-14	47.4	15.3	-
Cry1Fb	-	NODE_84_length_19137_cov_360.453	-	3369	3506	7715	7850	7694	7851	3507	+	3.5e-12	41.8	15.9	-
Cry1Jb	-	NODE_84_length_19137_cov_360.453	-	3380	3512	7718	7850	7697	7851	3513	+	1e-11	40.7	15.4	-
Cry1Ja	-	NODE_84_length_19137_cov_360.453	-	3366	3500	7713	7847	7691	7850	3503	+	1.4e-11	40.1	15.2	-
Cry1Be	-	NODE_84_length_19137_cov_360.453	-	3474	3605	7719	7850	7698	7851	3606	+	2.6e-11	39.2	15.2	-
Cry1Db	-	NODE_84_length_19137_cov_360.453	-	3344	3483	7714	7851	7694	7851	3483	+	5.2e-11	38.1	16.2	-
Cry1Jd	-	NODE_84_length_19137_cov_360.453	-	3369	3506	7713	7850	7692	7851	3507	+	7.2e-11	37.9	16.6	-
Cry9Ga	-	NODE_84_length_19137_cov_360.453	-	3549	3688	7711	7849	7690	7851	3690	+	7.7e-11	37.7	16.3	-
Cry1Da	-	NODE_84_length_19137_cov_360.453	-	3354	3490	7710	7846	7690	7848	3492	+	8.4e-11	37.4	15.3	-
Cry1Ka	-	NODE_84_length_19137_cov_360.453	-	3427	3569	7708	7850	7688	7851	3570	+	1.8e-10	36.3	16.0	-
Cry1Bb	-	NODE_84_length_19137_cov_360.453	-	3481	3607	7720	7846	7700	7848	3609	+	1.9e-10	36.4	14.1	-
Cry1Gb	-	NODE_84_length_19137_cov_360.453	-	3373	3510	7714	7851	7694	7851	3510	+	2.1e-10	36.0	15.3	-
Cry1Ai	-	NODE_84_length_19137_cov_360.453	-	3406	3545	7711	7850	7689	7851	3546	+	3.1e-10	35.3	15.8	-
Cry1Fa	-	NODE_84_length_19137_cov_360.453	-	3388	3524	7714	7850	7694	7851	3525	+	3.7e-10	35.4	15.1	-
Cry1Bf	-	NODE_84_length_19137_cov_360.453	-	3554	3686	7718	7850	7698	7851	3687	+	3.8e-10	35.5	15.3	-
Cry1Bi	-	NODE_84_length_19137_cov_360.453	-	3581	3713	7718	7850	7698	7851	3714	+	5.5e-10	34.9	15.3	-
Cry1Bg	-	NODE_84_length_19137_cov_360.453	-	3575	3707	7718	7850	7698	7851	3708	+	9.8e-10	34.1	15.2	-
Cry1Bj1	-	NODE_84_length_19137_cov_360.453	-	3569	3701	7718	7850	7698	7851	3702	+	1.2e-09	33.8	15.3	-
Cry1Ea	-	NODE_84_length_19137_cov_360.453	-	3379	3514	7714	7849	7694	7851	3516	+	1.7e-09	33.3	15.1	-
Cry1Ad	-	NODE_84_length_19137_cov_360.453	-	3403	3539	7714	7850	7694	7851	3540	+	2.4e-09	32.6	15.1	-
Cry1Ae	-	NODE_84_length_19137_cov_360.453	-	3409	3545	7714	7850	7693	7851	3546	+	3.8e-09	32.1	15.6	-
Cry1Eb	-	NODE_84_length_19137_cov_360.453	-	3395	3524	7721	7850	7701	7851	3525	+	3.9e-09	32.4	15.3	-
Cry1Bh	-	NODE_84_length_19137_cov_360.453	-	3636	3770	7716	7850	7696	7851	3771	+	7.6e-09	31.1	15.6	-
Cry1La	-	NODE_84_length_19137_cov_360.453	-	3376	3511	7714	7849	7692	7851	3513	+	8.5e-09	30.8	15.7	-

Figura 31 – Saída da execução do *nhmmscan* retirando parte das famílias de genes *cry* referente a seção A.3.

1	NODE_110_length_12556_cov_314.781	CRY_DETECT	gene	10454	11260	3.3e-101	+	.	Name= Cry30Aa	Bit-Score: 337.0	Bias: 46.8
2	NODE_118_length_10549_cov_579.939	CRY_DETECT	gene	287	2187	0	+	.	Name= Cry2Aa	Bit-Score: 2180.3	Bias: 121.6
3	NODE_118_length_10549_cov_579.939	CRY_DETECT	gene	7393	9577	0	-	.	Name= Cry1Ie	Bit-Score: 2233.6	Bias: 115.5
4	NODE_118_length_10549_cov_579.939	CRY_DETECT	gene	10062	10549	1.1e-145	-	.	Name= Cry1Fa	Bit-Score: 484.1	Bias: 18.1
5	NODE_11_length_133492_cov_140.468	CRY_DETECT	gene	109214	109417	0.23	-	.	Name= Cry30Aa	Bit-Score: 9.3	Bias: 23.0
6	NODE_124_length_8888_cov_128.481	CRY_DETECT	gene	5117	5304	0.011	-	.	Name= Cry30Aa	Bit-Score: 9.8	Bias: 18.6
7	NODE_166_length_3504_cov_159.262	CRY_DETECT	gene	2806	2904	0.23	-	.	Name= Cry30Aa	Bit-Score: 4.1	Bias: 17.7
8	NODE_172_length_3262_cov_120.933	CRY_DETECT	gene	2995	3168	0.22	-	.	Name= Cry7Gd	Bit-Score: 4.1	Bias: 14.6
9	NODE_186_length_2444_cov_1318.21	CRY_DETECT	gene	1	2444	0	+	.	Name= Cry1Ah	Bit-Score: 2579.0	Bias: 91.8
10	NODE_18_length_108561_cov_132.893	CRY_DETECT	gene	17405	17503	3.9	-	.	Name= Cry30Aa	Bit-Score: 4.9	Bias: 18.1
11	NODE_200_length_1638_cov_2092.64	CRY_DETECT	gene	1015	1638	4.5e-207	+	.	Name= Cry1Ai	Bit-Score: 684.8	Bias: 31.6
12	NODE_22_length_93611_cov_171.592	CRY_DETECT	gene	24508	24624	3.8	+	.	Name= Cry30Aa	Bit-Score: 4.8	Bias: 16.4
13	NODE_234_length_1013_cov_766.226	CRY_DETECT	gene	1	1013	1.9e-300	-	.	Name= Cry1Af	Bit-Score: 994.0	Bias: 41.5
14	NODE_235_length_1010_cov_839.119	CRY_DETECT	gene	1	1010	0	+	.	Name= Cry1Ai	Bit-Score: 1072.2	Bias: 32.5
15	NODE_250_length_872_cov_1616.63	CRY_DETECT	gene	1	872	2.4e-263	+	.	Name= Cry1A-like	Bit-Score: 870.5	Bias: 44.8
16	NODE_256_length_807_cov_903.879	CRY_DETECT	gene	1	506	8e-157	+	.	Name= Cry1Ai	Bit-Score: 517.1	Bias: 17.4
17	NODE_25_length_82350_cov_196.987	CRY_DETECT	gene	35045	35100	6.6	-	.	Name= Cry34Ac	Bit-Score: 9.3	Bias: 0.7
18	NODE_262_length_735_cov_883.235	CRY_DETECT	gene	1	735	5.9e-204	-	.	Name= Cry1Ca	Bit-Score: 673.5	Bias: 26.7
19	NODE_350_length_471_cov_1.02326	CRY_DETECT	gene	108	259	0.022	-	.	Name= Cry2Af	Bit-Score: 6.0	Bias: 9.5
20	NODE_358_length_466_cov_432.563	CRY_DETECT	gene	22	231	0.019	+	.	Name= Cry20Aa	Bit-Score: 5.9	Bias: 19.1
21	NODE_358_length_466_cov_432.563	CRY_DETECT	gene	309	449	0.15	+	.	Name= Cry20Aa	Bit-Score: 3.0	Bias: 13.8
22	NODE_37_length_48629_cov_194.03	CRY_DETECT	gene	16674	16776	1.4	+	.	Name= Cry29Aa	Bit-Score: 7.1	Bias: 5.1
23	NODE_4_length_185155_cov_190.951	CRY_DETECT	gene	141452	141519	3.1	+	.	Name= Cry34Aa	Bit-Score: 11.5	Bias: 3.3
24	NODE_59_length_27247_cov_379.006	CRY_DETECT	gene	22373	22847	1.9e-41	-	.	Name= Cry73Aa	Bit-Score: 140.7	Bias: 49.9
25	NODE_606_length_380_cov_4156.68	CRY_DETECT	gene	1	380	6.6e-120	-	.	Name= Cry1Ga	Bit-Score: 393.6	Bias: 7.8
26	NODE_61_length_25976_cov_143.362	CRY_DETECT	gene	3164	3357	7.8e-06	+	.	Name= Cry46Aa	Bit-Score: 24.8	Bias: 10.5
27	NODE_640_length_245_cov_986.542	CRY_DETECT	gene	1	245	3.9e-75	-	.	Name= Cry1Ga	Bit-Score: 244.4	Bias: 10.8
28	NODE_655_length_158_cov_439.774	CRY_DETECT	gene	17	143	0.21	-	.	Name= Cry20Aa	Bit-Score: 0.9	Bias: 10.2
29	NODE_657_length_154_cov_44.0741	CRY_DETECT	gene	20	140	1	-	.	Name= Cry20Aa	Bit-Score: -1.4	Bias: 10.9
30	NODE_659_length_133_cov_839.833	CRY_DETECT	gene	24	129	5.7	-	.	Name= Cry20Aa	Bit-Score: -4.0	Bias: 9.6
31	NODE_67_length_23434_cov_215.945	CRY_DETECT	gene	23260	23325	0.11	-	.	Name= Cry60Aa	Bit-Score: 11.5	Bias: 2.4
32	NODE_71_length_22379_cov_167.84	CRY_DETECT	gene	13063	13134	5.4	+	.	Name= Cry64Ba	Bit-Score: 5.8	Bias: 2.7
33	NODE_72_length_22303_cov_155.942	CRY_DETECT	gene	21264	21333	3.4	+	.	Name= Cry34Ab	Bit-Score: 8.7	Bias: 3.8
34	NODE_7_length_146872_cov_132.479	CRY_DETECT	gene	9162	9290	0.52	+	.	Name= Cry35Ba	Bit-Score: 11.2	Bias: 9.1
35	NODE_7_length_146872_cov_132.479	CRY_DETECT	gene	51390	51506	1.3	-	.	Name= Cry34Ba	Bit-Score: 12.6	Bias: 7.9
36	NODE_84_length_19137_cov_360.453	CRY_DETECT	gene	7708	7853	8.6e-14	+	.	Name= Cry1Cb	Bit-Score: 47.4	Bias: 15.3
37	NODE_84_length_19137_cov_360.453	CRY_DETECT	gene	10288	12188	0	+	.	Name= Cry2Ab	Bit-Score: 2172.2	Bias: 131.0
38	NODE_87_length_18294_cov_409.757	CRY_DETECT	gene	1434	1632	1.2e-28	+	.	Name= Cry30Aa	Bit-Score: 96.9	Bias: 12.1
39	NODE_87_length_18294_cov_409.757	CRY_DETECT	gene	15178	15719	2.1e-47	+	.	Name= Cry30Aa	Bit-Score: 159.1	Bias: 30.9
40	NODE_94_length_15414_cov_99.2846	CRY_DETECT	gene	6355	6662	6.5e-05	+	.	Name= Cry22Bb	Bit-Score: 21.3	Bias: 16.4
41	NODE_95_length_15200_cov_205.991	CRY_DETECT	gene	6134	6244	0.092	+	.	Name= Cry5Ad	Bit-Score: 8.0	Bias: 7.4
42	NODE_96_length_15085_cov_216.286	CRY_DETECT	gene	12514	12611	1.8	-	.	Name= Cry34Aa	Bit-Score: 8.7	Bias: 4.5
43	NODE_9_length_135708_cov_151.664	CRY_DETECT	gene	103870	104081	5.8	-	.	Name= Cry1Bj1	Bit-Score: 4.5	Bias: 10.8

Figura 32 – Informações do genes *cry* retirando parte das famílias de genes *cry* referente a seção A.3.