Universidade de Ribeirão Preto - UNAERP
Departamento de Biotecnologia
Programa de Pós-Graduação em Biotecnologia

David Buzatto

Especificidade de Proteínas Cry de *Bacillus thuringiensis* Baseada nas Características Conformacionais de Suas Estruturas Terciárias

Ribeirão Preto
2017

David Buzatto

Especificidade de Proteínas Cry de *Bacillus thuringiensis* Baseada nas Características Conformacionais de Suas Estruturas Terciárias

Tese apresentada ao Programa de Pós-Graduação *Stricto Sensu* da Universidade de Ribeirão Preto - UNAERP, como requisito parcial para a obtenção do título de Doutor em Biotecnologia.

Orientadora: Profa. Dra. Sonia Marli Zingaretti.

Ribeirão Preto

2017

Ficha catalográfica preparada pelo Centro de Processamento Técnico
da Biblioteca Central da UNAERP

- Universidade de Ribeirão Preto -

Buzatto, David, 1985-
B992e Especificidade de Proteínas Cry de *Bacillus thuringiensis* Baseada nas características conformacionais de suas estruturas terciárias / David Buzatto. - - Ribeirão Preto, 2017.
226 f.: il. color.

Orientadora: Profª. Drª. Sonia Marli Zingaretti

Tese (doutorado) - Universidade de Ribeirão Preto, UNAERP, Biotecnologia. Ribeirão Preto, 2017.

1. *Bacillus thuringiensis*. 2. Proteínas Cry. 3. Genes Cry. 4. Proteínas – Comparação. I. Título.

CDD 660

DAVID BUZATTO

ESPECIFICIDADE DE PROTEÍNAS CRY DE *BACILLUS THURINGIENSIS* BASEADA NAS CARACTERÍSTICAS CONFORMACIONAIS DE SUAS ESTRUTURAS TERCIÁRIAS

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Biotecnologia da Universidade de Ribeirão Preto, para obtenção do título de Doutor em Biotecnologia.

Área de Concentração: Biotecnologia Aplicada à Saúde

Data da defesa: 18 de dezembro de 2017

Resultado: Aprovado

BANCA EXAMINADORA

Profa. Dra. Sonia Marli Zingaretti
Universidade de Ribeirão Preto - UNAERP

Profa. Dra. Suzelei de Castro França
Universidade de Ribeirão Preto - UNAERP

Prof. Dr. Edilson Carlos Caritá
Universidade de Ribeirão Preto – UNAERP

Prof. Dr. Nilson Nicolau Junior
Universidade Federal de Uberlândia

Prof. Dr. Eduardo Marmo Moreira
Instituto Federal de São Paulo - IFSP

RIBEIRÃO PRETO
2017

À minha esposa Fernanda,
à minha mãe Selma,
ao meu irmão Pedro e
aos meus grandes amigos:
Breno, Eduardo, Everton, Luiz,
Gustavo, Rodrigo e Vágner.

AGRADECIMENTOS

Agradeço a todos que me ajudaram, direta ou indiretamente, no desenvolvimento deste trabalho, em especial, à minha esposa Fernanda e à minha orientadora, Profa. Sonia. Agradeço também ao apoio financeiro oferecido pelo Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) no custeio de mensalidades do programa de pós-graduação em biotecnologia e pela Universidade de Ribeirão Preto (UNAERP) no auxílio no pagamento da publicação de artigos. Muito obrigado!

"*Any fool can write code that a computer can understand. Good programmers write code that humans can understand*".

Martin Fowler

RESUMO

A utilização de produtos químicos de forma inconsciente e indiscriminada tem contaminado, a cada dia mais, o planeta Terra. Uma das alternativas para a diminuição desta poluição é a utilização de biopesticidas, sendo que a utilização do *Bacillus thuringiensis*, uma bactéria encontrada no solo, tem sido feita há pelo menos cinquenta anos com este objetivo, pois a mesma sintetiza um conjunto de proteínas, chamadas de proteínas Cry, que são tóxicas a diversas ordens de insetos, a alguns tipos de ácaros, a nematoides e a células cancerígenas. Neste trabalho, fazendo o uso de algoritmos de alinhamento/comparação estrutural de proteínas, busca-se identificar quais seriam as modificações da conformação dessas proteínas que influenciam a especificidade para cada ordem de inseto, contribuindo no desenvolvimento e/ou descoberta de proteínas bioinseticidas mais eficazes no controle de diferentes pragas, além da utilização dos genes codificadores dessas proteínas em plantas transgênicas. Após o processamento dos dados obtidos no experimento delineado, foi possível verificar que existem modificações conformacionais nas estruturas das proteínas analisadas que provavelmente têm impacto na especificidade dessas proteínas nos insetos alvo de algumas ordens, sendo a Volta 2 da proteína Cry1Aa1 importante para a manifestação contra *Lepidoptera* de forma geral e a algumas espécies de *Diptera*, o *Loop* 1 da proteína Cry3Aa1 importante para a atividade contra *Coleoptera* e os resíduos ^{157}R^{159}, ^{169}Y^{171}, ^{242}W^{244}, ^{245}F^{247}, ^{248}Y^{250} e ^{263}F^{265} de Cry4Ba1 são importantes para a atividade em todas as proteínas comparadas, visto a alta conservação apresentada. Como neste trabalho foi utilizada uma abordagem *in silico* para realizar essa verificação, faz-se necessário, como trabalho futuro, a execução de experimentos *in vivo* para a confirmação de tais resultados. Foram também desenvolvidos dois *softwares* durante o ciclo de vida desta pesquisa, sendo eles o "CryGetter", um aplicativo capaz de consolidar dados das proteínas Cry a partir de duas fontes de dados e o "*3-Domain Cry Protein Models Comparison Lab*", utilizado para apresentar visualmente os resultados gerados.

Palavras-chave: *Bacillus thuringiensis*. Proteínas Cry. Genes Cry. Comparação Estrutural de Proteínas. Alinhamento Estrutural de Proteínas.

ABSTRACT

The unconsciously and indiscriminate use of chemicals has contaminated, every day more, the planet Earth. One alternative to reduce this pollution is the use of biopesticides, wherein the use of Bacillus thuringiensis, a bacterium found in soil, has been made for at least fifty years with this goal, because it synthesizes a set of proteins, called Cry proteins, that are toxic to various orders of insects, some types of mites and nematoides. In this work, making the use of protein structural alignment/comparison algorithms, the goal is to identify what are the conformation modifications of these proteins that influence the specificity for each insect order, contributing to the development and/or discovery of new biopesticidal proteins that are more effective in different pest control, in addition to the use of these protein-coding genes in transgenic plants. After processing the data obtained in the delineated experiment, it was possible to verify that there are conformational modifications in the structures of the analyzed proteins that probably have impact on the specificity of these proteins in the insects target of some orders, being the Turn 2 of the Cry1Aa1 protein important for a manifestation against *Lepidoptera* in general way and to some species of *Diptera*, the Loop 1 of the Cry3Aa1 protein is important for an activity against *Coleoptera* and the residues ^{157}R^{159}, ^{169}Y^{171}, ^{242}W^{244}, ^{245}F^{247}, ^{248}Y^{250} and ^{263}F^{265} of Cry4Ba1 are important for an activity on all compared proteins, due to the high conservation presented. As in this work an *in silico* approach was used to carry out this verification, it is necessary, as future work, to carry out *in vivo* experiments for the confirmation of such results. Also, two softwares were developed during the life cycle of this research, being the "CryGetter" an application capable of consolidating data of the Cry proteins from two data sources and the "3-Domain Cry Protein Models Comparison Lab", used to visual display the generated results.

Keywords: *Bacillus thuringiensis*. Cry Proteins. Cry Genes. Protein Structural Comparison. Protein Structural Alignment.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

1D	Uma Dimensão
2D	Duas Dimensões
3D	Três Dimensões
ABC	*ATP-Binding Cassette*
AFP	*Aligned Fragment Pair*
API	*Application Programming Interface*
ATP	Adenosina Trifosfato
BBM	*Brush Border Membrane*
BLOSUM	*BLOcks SUbstitution Matrix*
CAB-align	*Contact Area-Based Alignment*
CATH	*Class Architecture Topology Homologous superfamily*
CE	*Combinatorial Extension*
Cry	*Crystal Protein*
Cyt	*Cytolytic Protein*
DNA	*Deoxyribonucleic Acid*
EBS	*Entropy Balanced Statistical*
EBI	*European Bioinformatics Institute*
ePC	*efficient enumeration-based Protein structure Comparison*
FASE	*Flexible Alignment of Secondary structure Elements*
FATCAT	*Flexible structure AlignmenT by Chaining AFPs with Twists*
FLASH	*Fast aLignment Algorithm for finding Structural Homology of proteins*
FSN	*Flexible Structural Neighborhood*
FORTRAN	IBM *Mathematical FORmula TRANslation System*

GOSSIP	*GlObal Structure SuperposItion of Proteins*
HTML	*Hypertext Markup Language*
HTTPS	*Hypertext Transfer Protocol Secure*
iPBA	*Improved Protein Block Alignment*
MAMMOTH	*MAtching Molecular Models Obtained from THeory*
MATRAS	*MArkov TRAnsition of protein Structure evolution*
MMDB	*Molecular Modeling Database*
MOMA	*MOrphing & MAtching*
NCBI	*National Center for Biotechnology Information*
NP-Hard	*Nondeterministic Polynomial time Hard*
PADS	*Protein Alignment by Directional shape Signatures*
PAM	*Point Accepted Mutations*
PB	*Protein Block*
PD	Programação Dinâmica
PDB	*Protein Data Bank*
PMDB	*Protein Model Database*
PPM	*Phenotypic Plasticity Method*
PROuST	*PROtein STructure comparison*
RMN	Ressonância Magnética Nuclear
RMSD	*Root-Mean-Square Distance* ou *Root-Mean-Square Deviation*
RNA	*Ribonucleic Acid*
RSRZ	*R-value Z-score*
URMSD	*Unit-Vector Root-Mean-Square Distance*
SAP	*Structure Alignment Program*
SAT	*Shape And Transformation*
SCALE	*Structure-Conscious ALignment of secondary structure Elements*

SCOP	*Structural Classification of Proteins*
Sip	*Secreted Insecticidal Protein*
SSE	*Secondary Structure Element*
SSEF	*Secondary Structure Element Footprint*
STAMP	*STructual Alignment of Multiple Proteins*
STRUCLA	*STRUcture CLAssification*
SEGA	*Semiglobal Graph Alignment*
TAD	Tipo Abstrato de Dados
TM-Align	*Template/Model-Align*
TM-Align	*Template/Model-Score*
TS-AMIR	*Topology String Alignment Method for Intensive Rapid comparison of protein structures*
UML	*Unified Modeling Language*
URL	*Uniform Resource Locator*
VAST	*Vector Alignment Search Tool*
Vip	*Vegetative Insecticidal Protein*
VMD	*Visual Molecular Dynamics*
XLSX	Microsoft Excel *Open* XML *Format Spreadsheet*
XML	*Extensible Markup Language*

LISTA DE SÍMBOLOS

α	Letra grega minúscula Alfa
β	Letra grega minúscula Beta
δ	Letra grega minúscula Delta
Δ	Letra grega maiúscula Delta
θ	Letra grega minúscula Teta
γ	Letra grega minúscula Gama
λ	Letra grega minúscula Lambda
μ	Letra grega minúscula Mi
π	Letra grega minúscula Pi
ρ	Letra grega minúscula Rô
σ	Letra grega minúscula Sigma
Σ	Letra grega maiúscula Sigma (somatório)
ϕ	Letra grega minúscula Fi
ψ	Letra grega minúscula Psi
ω	Letra grega minúscula Ômega
Å	Ångström, unidade de medida de comprimento, sendo que $1Å = 10^{-10}m$

SUMÁRIO

1 INTRODUÇÃO

A utilização recorrente de produtos químicos nocivos ao meio ambiente, bem como o descarte indiscriminado de agentes poluidores em rios e no solo, contaminando os lençóis freáticos, vem se tornando a cada dia um problema maior. As autoridades municipais, estaduais e federais têm se mobilizado para controlar esse tipo de ação, visto que a poluição ambiental influencia negativamente toda a sociedade e a biosfera, dificultando a administração por parte dos governos, onerando a máquina pública e piorando a qualidade de vida da população, além de degradar os sistemas biológicos existentes no planeta.

Dentre os produtos químicos nocivos que são descartados inapropriadamente ou que apresentam efeitos colaterais em decorrência do seu uso, estão os pesticidas, que embora sejam usados para resolver o problema das pragas nas lavouras, possibilitando o aumento da produção de alimentos, podem acabar contaminando o solo e os lençóis freáticos com o passar do tempo, trazendo mais problemas para toda a biosfera terrestre, incluindo a população humana, do que bons resultados a longo prazo.

Uma alternativa à utilização de pesticidas químicos é a utilização de biopesticidas, que utilizam agentes biológicos no controle de pragas da lavoura ou de vetores de doenças humanas ou de outros animais. Esse tipo de pesticida se mostra viável, pois é mais fácil de ser produzido e agride pouco ou nada o meio ambiente. Um dos micro-organismos que são utilizados na manufatura de biopesticidas é o *Bacillus thuringiensis* (*Bt*), uma bactéria que tem como produto de seu metabolismo, durante o processo de esporulação, um conjunto de proteínas tóxicas a diversas ordens de insetos. Estas proteínas, em contato com o intestino dos insetos, tem ação entomopatogênica, criando poros nas paredes intestinais, degradando assim o intestino, influenciando negativamente no processo digestivo do organismo, causando inanição e contaminação por bactérias previamente instaladas no inseto, consequentemente sendo responsável por sua morte.

Dentre os diversos tipos de proteínas sintetizadas pelo *Bt* estão as proteínas cristal (*Crystal Protein*), ou proteínas Cry. Como dito, a toxicidade da proteína é expressa apenas no processo de digestão dos insetos, não sendo tóxicas para a grande maioria dos outros seres vivos. As proteínas Cry são divididas em diversas classes, onde cada classe é tóxica para determinadas ordens de insetos, além de também existirem algumas variações que são tóxicas a ácaros e a nematoides.

Além da sintetização desses compostos em escala industrial, pode-se também criar plantas transgênicas que têm o gene cry do bacilo inserido em suas células, permitindo que as plantas modificadas sintetizem as proteínas Cry, contaminando os insetos que se alimentarem delas.

Pelo exposto, pode-se notar que a utilização do *Bt* para a criação de biopesticidas e plantas transgênicas é viável, sendo assim, essa bactéria tem sido alvo de vários estudos.

2 REVISÃO DA LITERATURA

2.1 PROTEÍNAS

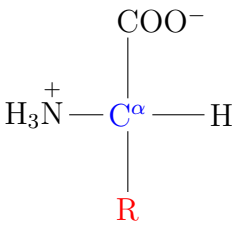
Segundo Lesk (2016), as proteínas compreendem um conjunto de macromoléculas biológicas que, em seu estado nativo, ou seja, após serem sintetizadas e estarem imersas, por exemplo, no citoplasma, adquirem uma estrutura tridimensional exclusiva a qual lhes confere suas funções específicas. Lesk (2016) categoriza as proteínas em sete classes:

- **Estruturais:** como o próprio nome diz, são proteínas que são usadas pelos organismos com o objetivo de criar estruturas, por exemplo, a queratina presente na pele, pelos, unhas, etc. dos mamíferos;
- **Enzimas:** são catalizadores do metabolismo e dos processos de replicação e transcrição do *Deoxyribonucleic Acid* (DNA);
- **Anticorpos:** atuam como reconhecedores e no mecanismo de expulsão de agentes patogênicos invasores;
- **Reguladoras:** atuam como agentes reguladores dos processos celulares, por exemplo, a regulação de quais genes devem ser transcritos em um determinado momento;
- **Sensores:** têm papel de agirem como transmissores de sinais, por exemplo, sinais internos de um organismo, como uma dor em algum órgão, ou então de sinais percebidos no ambiente, como a variação de temperatura;
- **Transportadoras e Bombas:** controlam o tráfego de entrada e saída de solutos em uma célula;
- **Transdutoras:** são as proteínas capazes de converter energia química em mecânica, por exemplo, na contração dos músculos e também as Adenosina Trifosfato (ATP) sintases, que convertem energia quimiosmótica obtida na respiração em energia química que é armazenada nas ligações fosfato dos ATP;

As proteínas são compostas, de acordo com Nelson e Cox (2014), por subunidades monoméricas simples, denominadas aminoácidos, sendo que todas as proteínas são construídas utilizando um mesmo conjunto de vinte aminoácidos. As proteínas são definidas então, como polímeros de aminoácidos, sendo que cada resíduo[1] de aminoácido que as compõe

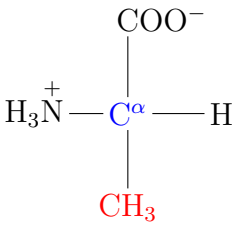
[1] Resíduo, ou resíduo de aminoácido, no contexto das proteínas, indica que houve uma perda de uma molécula de água para cada ligação feita entre dois aminoácidos (NELSON; COX, 2014).

está ligado ao(s) seu(s) vizinho(s) por meio de uma ligação covalente denominada ligação peptídica. Ainda, segundo Nelson e Cox (2014), os vinte aminoácidos são classificados como α-aminoácidos, pois possuem um grupo carboxil e um grupo amino ligados ao um mesmo átomo de carbono, denominado C^{α} (α), diferindo um do outro por causa de suas respectivas cadeias laterais, chamadas de grupos R, também ligadas ao C^{α}. A estrutura geral de um α-aminoácido, com exceção da Prolina, pode ser vista na Fórmula Estrutural 2.1, estando o C^{α} colorido em azul e a posição do grupo R colorida em vermelho. Na Fórmula Estrutural 2.2 pode ser visto o aminoácido Alanina, que possui o grupo R CH_3. O aminoácido Prolina, apresentado na Fórmula Estrutural 2.3, é uma exceção, pois possui uma cadeia alifática com uma estrutura cíclica.



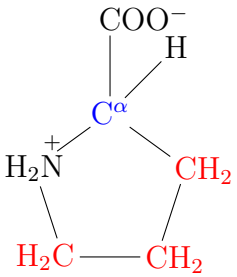
(2.1)

α-aminoácido



(2.2)

Alanina



(2.3)

Prolina

Existem ainda vários outros aminoácidos menos comuns além desses vinte, sendo que alguns são modificações pós-translacionais dos mesmos, além de aminoácidos que estão presentes nos organismos, mas que não são usados na síntese das proteínas (NELSON; COX, 2014). Na Tabela 1 são apresentados os vinte aminoácidos em conjunto com suas respectivas abreviações e símbolos.

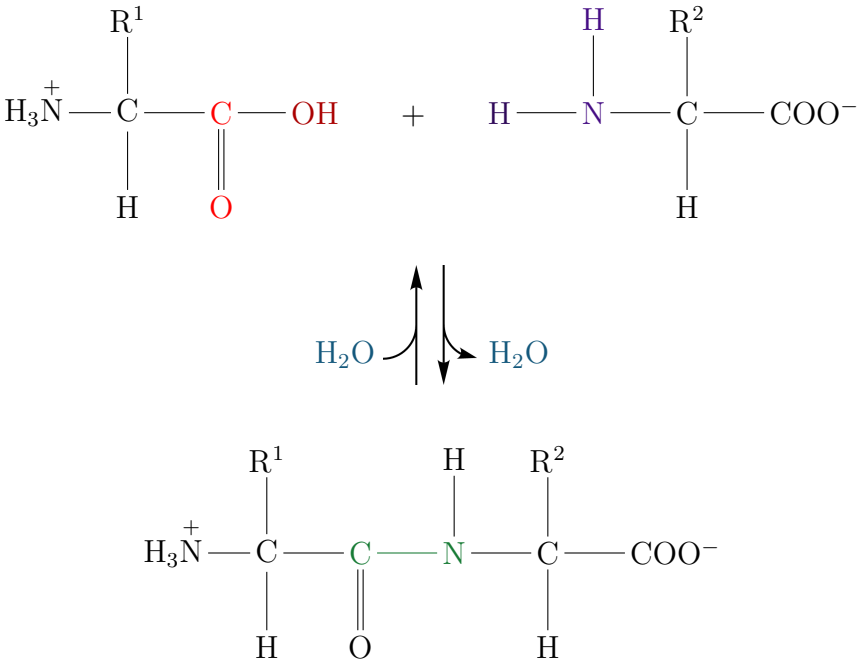
Tabela 1 – Os vinte aminoácidos constituintes das proteínas

Aminoácido	**Abreviação**	**Símbolo**
Grupos R alifáticos, apolares		
Glicina	Gly	G
Alanina	Ala	A
Prolina	Pro	P
Valina	Val	V
Leucina	Leu	L
Isoleucina	Ile	I
Metionina	Met	M
Grupos R aromáticos		
Fenilalanina	Phe	F
Tirosina	Tyr	Y
Triptofano	Trp	W
Grupos R polares, não carregados		
Serina	Ser	S
Treonina	Thr	T
Cisteína	Cys	C
Asparagina	Asn	N
Glutamina	Gln	Q
Grupos R carregados positivamente		
Lisina	Lys	K
Histidina	His	H
Arginina	Arg	R
Grupos R carregados negativamente		
Aspartato	Asp	D
Glutamato	Glu	E

Fonte: Adaptado de Nelson e Cox (2014) pelo autor

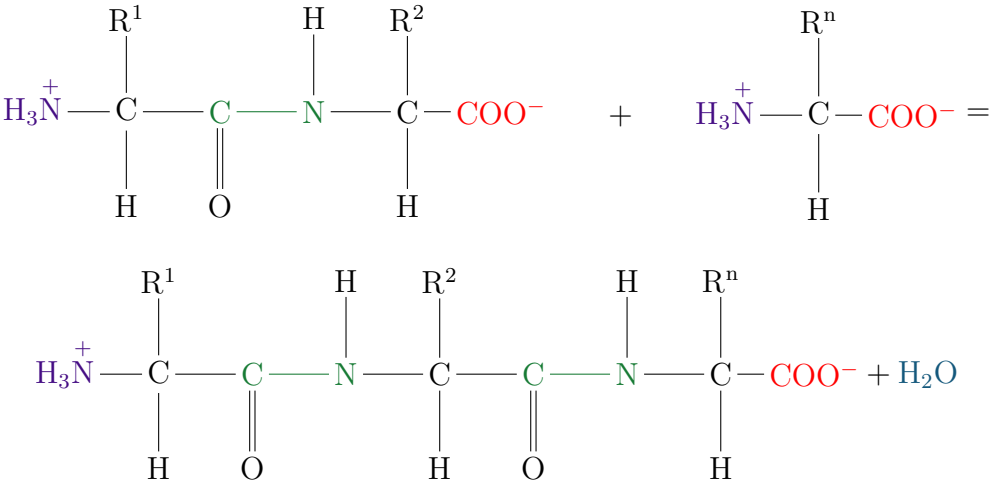
Durante o processo de síntese de uma proteína, como já dito, os aminoácidos são ligados uns aos outros por meio de ligações peptídicas, sendo que na Fórmula Estrutural 2.4 pode-se ver um esquema geral dessa reação, que ao acontecer, gera como resultado um dipeptídeo e uma molécula de água. Na Fórmula Estrutural 2.4, na primeira linha, que contém os dois aminoácidos envolvidos na reação, o grupo carboxil do primeiro aminoácido está colorido em vermelho e o grupo amino do segundo está colorido em roxo. A molécula de água obtida ao se condensar os dois aminoácidos originais para criar o dipeptídeo, ou a molécula de água usada na hidrolisação do dipeptídeo em dois aminoácidos, estão

destacadas em azul e, no dipeptídeo, a ligação peptídica está destacada em verde.



(2.4)

Vale ainda ressaltar que, segundo Lesk (2016), as proteínas são sintetizadas sempre ligando-se a extremidade c-terminal, ou seja, a extremidade que possui o grupo carboxil, com a extremidade n-terminal, do grupo amina, do próximo aminoácido que será condensado na cadeia de aminoácidos da proteína. Na Fórmula Estrutural 2.5 pode-se ver o esquema geral dessa reação, utilizando as cores roxo, vermelho, verde e azul para representar, respetivamente, a extremidade n-terminal, a extremidade c-terminal, a ligação peptídica e a molécula de água.

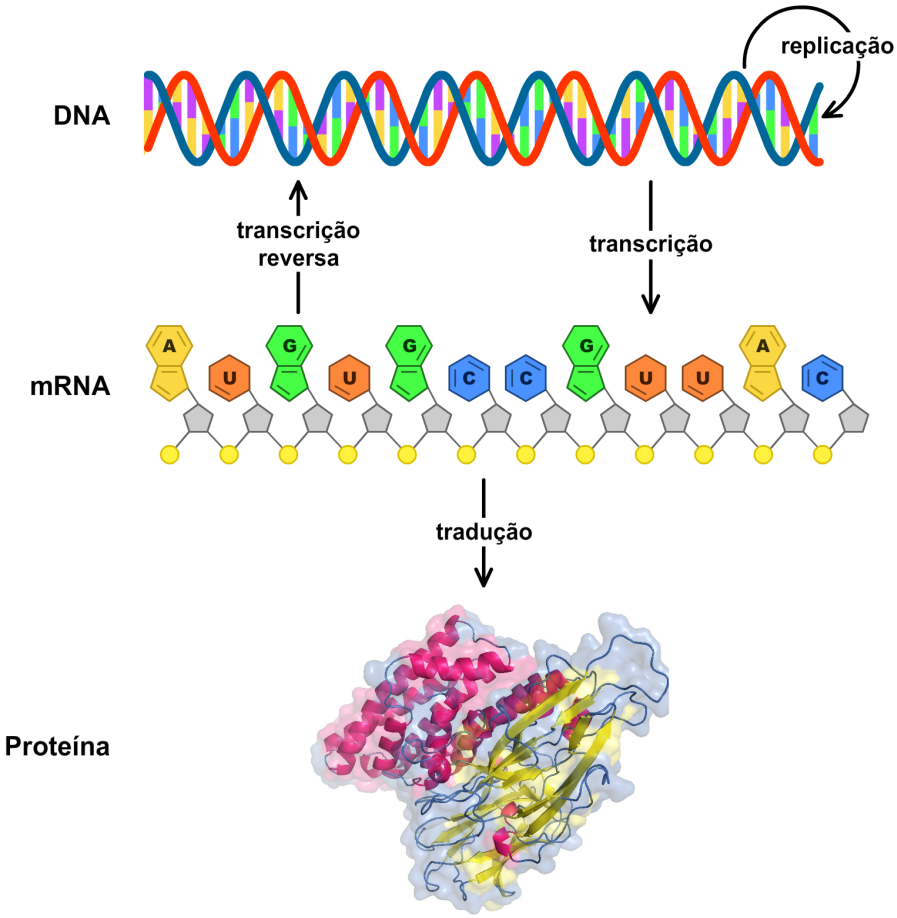


(2.5)

O processo de síntese das proteínas, segundo Lesk (2016), obedece ao dogma central da biologia molecular, proposto por Francis Crick em 1958 (CRICK, 1958; CRICK, 1970). Um diagrama representativo do dogma central expandido, aceito atualmente, é apresentado

na Figura 1, visto que na época da proposta do modelo original de Crick, ainda não se conhecia o mecanismo de transcrição reversa (LESK, 2016).

Figura 1 – Dogma Central da Biologia Molecular



Fonte: Adaptada de Lesk (2016) pelo autor

Pode-se verificar que, a partir do diagrama apresentado na Figura 1, uma fita da molécula de DNA é usada no processo de transcrição como molde para se obter uma molécula de *Ribonucleic Acid* (RNA) mensageiro (mRNA), que por sua vez, através do processo de tradução, é usada para sintetizar uma proteína, que adota sua forma nativa no fim desse processo, acarretando, consequentemente, na definição de sua função (LESK, 2016). Além disso, nota-se que o DNA pode ser replicado, ou seja, podem ser sintetizadas cópias da molécula de DNA, além de o RNA poder ser transcrito de forma reversa em uma molécula de DNA.

De forma mais detalhada, a síntese das proteínas, de acordo com Lesk (2016), inicia-se a partir da transcrição de um gene, definido como um segmento de DNA, em uma molécula de RNA. Ao passo que em procariontes os genes são sequências contíguas do DNA e são transcritos em RNA mensageiros diretamente, em eucariontes os genes

são divididos em subsequências denominadas éxons[2] e íntrons[3] que são transcritos em moléculas de RNA, as quais ainda precisam ser processadas para a remoção dos íntrons, inserção do capacete Cap'5 na extremidade 5' e da cauda Poli-A na extremidade 3' da molécula modificada, gerando assim, as moléculas finais de RNA mensageiro, as quais contém apenas os éxons e os componentes das extremidades (LESK, 2016).

O RNA mensageiro, por sua vez, é processado pelo ribossomo, uma organela celular composta de duas subunidades, denominadas subunidade maior e subunidade menor. O ribossomo é capaz de ler a fita de RNA mensageiro, que passa entre as subunidades, um códon[4] por vez, e, a partir da interpretação de cada códon, baseado no código genético do organismo em questão (Tabela 2), polimerizar/condensar um novo aminoácido na molécula de proteína que está sendo sintetizada, onde cada novo aminoácido é transportado ao ribossomo por uma molécula de RNA transportador (tRNA) específica, visto que é na estrutura do tRNA que existe o anticódon que pareia exatamente com o códon que está sendo traduzido (LESK, 2016). Na Figura 2 é apresentado graficamente esse processo.

Tabela 2 – O código genético padrão em códons de RNA

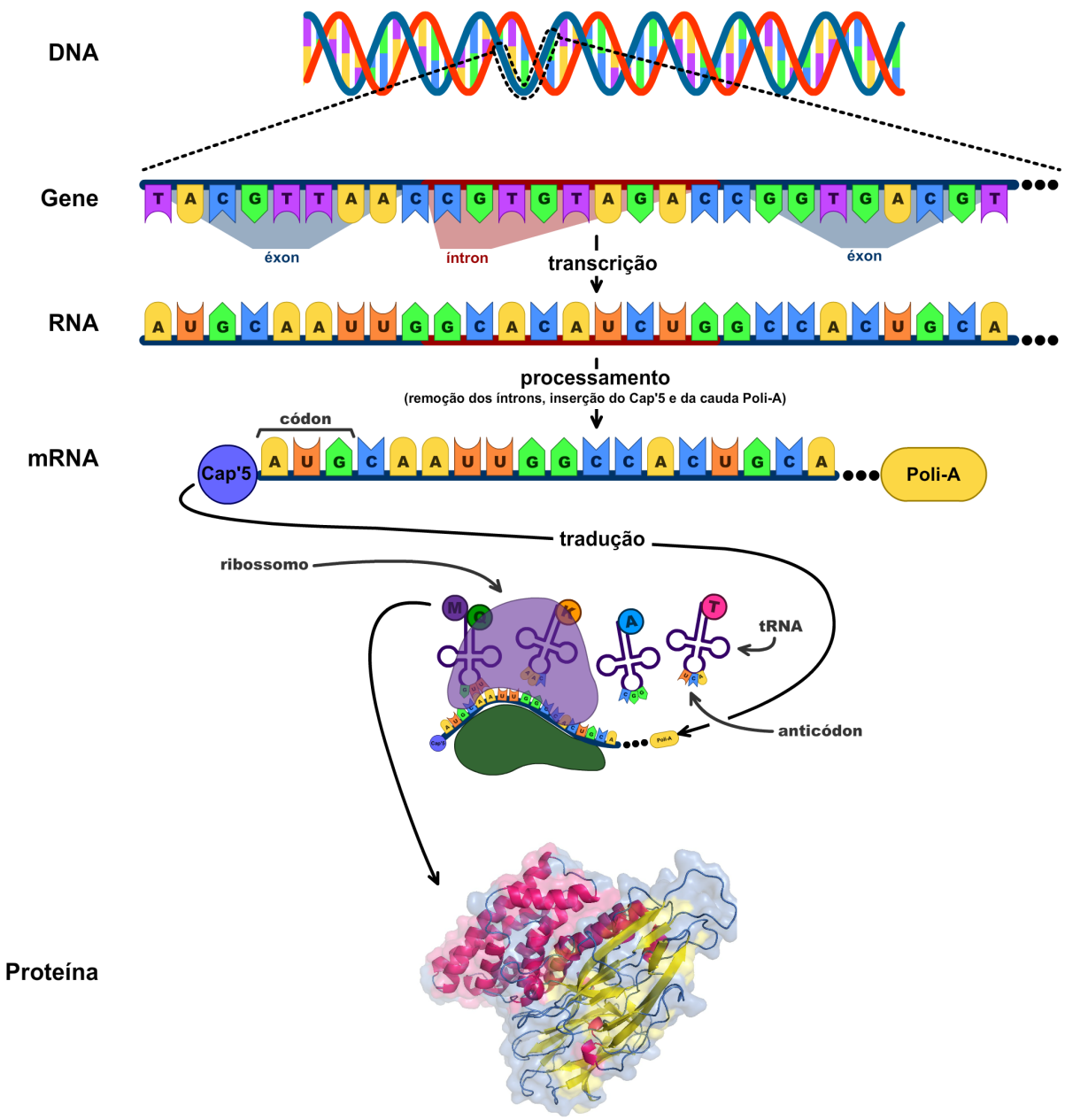
Códon	**Aa****	**Códon**	**Aa**	**Códon**	**Aa**	**Códon**	**Aa**
UUU	F	UCU	S	UAU	Y	UGU	C
UUC	F	UCC	S	UAC	Y	UGC	C
UUA	L	UCA	S	UAA	*	UGA	*
UUG	L	UCG	S	UAG	*	UGG	W
CUU	L	CCU	P	CAU	H	CGU	R
CUC	L	CCC	P	CAC	H	CGC	R
CUA	L	CCA	P	CAA	Q	CGA	R
CUG	L	CCG	P	CAG	Q	CGG	R
AUU	I	ACU	T	AAU	N	AGU	S
AUC	I	ACC	T	AAC	N	AGC	S
AUA	I	ACA	T	AAA	K	AGA	R
AUG	M	ACG	T	AAG	K	AGG	R
GUU	V	GCU	A	GAU	D	GGU	G
GUC	V	GCC	A	GAC	D	GGC	G
GUA	V	GCA	A	GAA	E	GGA	G
GUG	V	GCG	A	GAG	E	GGG	G

*códon de parada; **aminoácido.

Fonte: Adaptado de Lesk (2016) pelo autor

[2] Tradução de *exon* (*expressed region*)
[3] Tradução de *intron* (*intervening region*)
[4] Tradução de *codon* (*coding unit*), unidade de codificação, formada por um conjunto de três ribonucleotídeos.

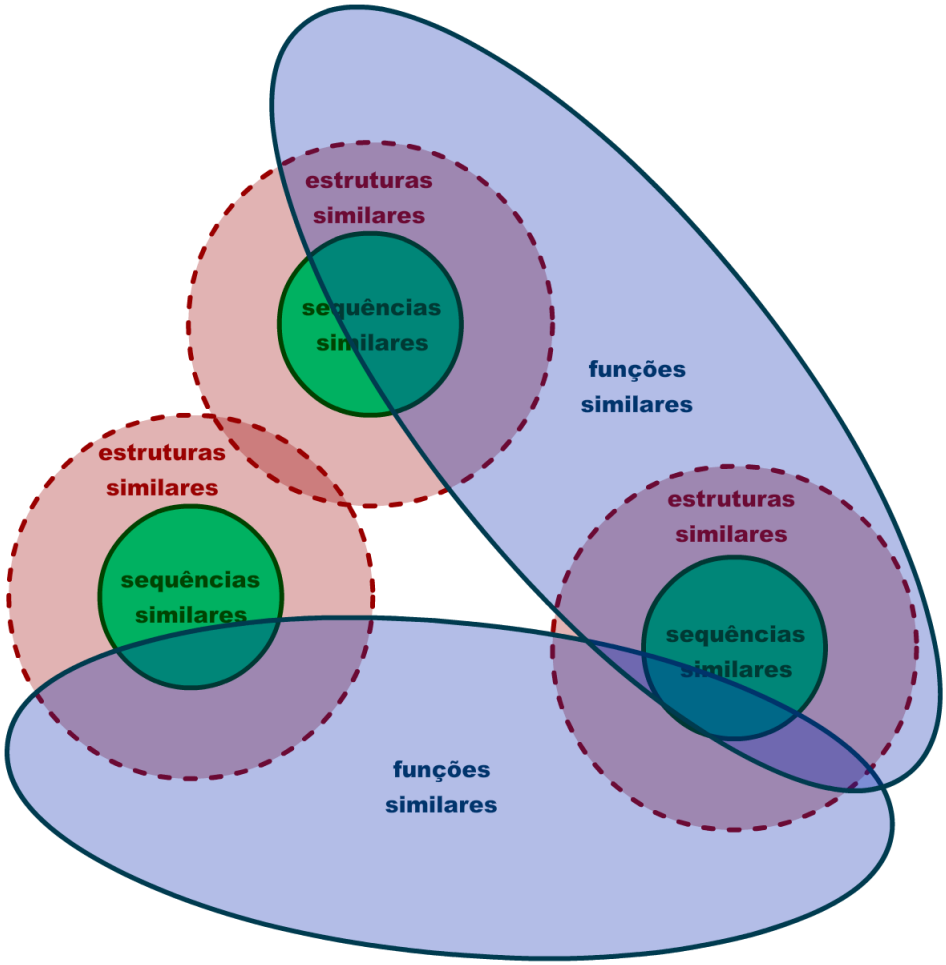
Figura 2 – Do gene à proteína



Fonte: Elaborada pelo autor

Após o processo de tradução, a proteína se enovela, ou se conforma, a partir da interação com o meio e com toda a sua cadeia, tomando assim, sua forma tridimensional padrão, ou seja, adotando seu estado nativo. No que diz respeito ao relacionamento entre proteínas diferentes, no que toca suas sequências de resíduos, estruturas nativas manifestadas e funções, Lesk (2016) traça um panorama importante, apresentado no diagrama de Veen da Figura 3 e explicado na sequência.

Figura 3 – Relações entre sequência, estrutura e função em diferentes proteínas



Fonte: Adaptada de Lesk (2016) pelo autor

A partir do diagrama da Figura 3, segundo Lesk (2016), pode-se afirmar que:

- Sequências similares normalmente produzem estruturas tridimensionais similares, sendo que esta similaridade diminui progressivamente ao passo que a similaridade das sequências diminui;

- De forma inversa, encontra-se também estruturas similares entre duas ou mais proteínas que possuem sequências de resíduos pouco similares, fazendo com que em várias situações, a classificação das proteínas de acordo com uma família de proteínas

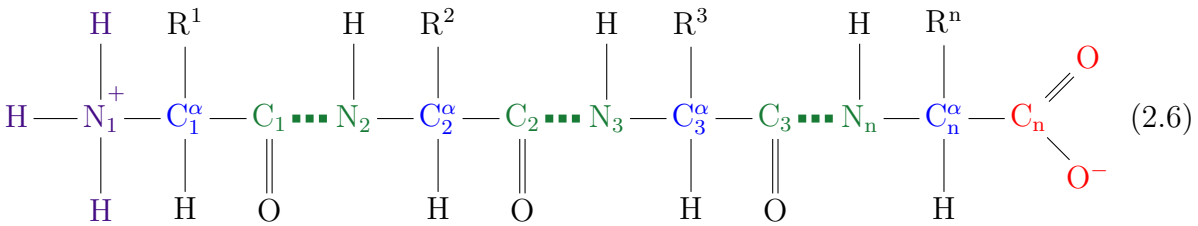
comum, só seja possível a partir da verificação da similaridade estrutural ao invés da similaridade de sequência;

- Sequências e estruturas similares implicam, normalmente, em funções análogas;
- Novamente, de forma inversa, existem também casos de proteínas com funções similares, mas que não apresentam homologia, ou seja, que não têm relação evolutiva entre suas sequências de aminoácidos, nem similaridade estrutural.

Pelo exposto, pode-se afirmar que uma proteína tem função similar ou igual à outra, na maioria das vezes, a partir da verificação da similaridade de suas estruturas. Visto a importância da estrutura tridimensional das proteínas, a seguir será formalizada a organização estrutural das proteínas, proposta por K. U. Linderstrøm-Lang e expandida por J. D. Bernal, segundo Lesk (2016).

2.1.1 Organização Estrutural

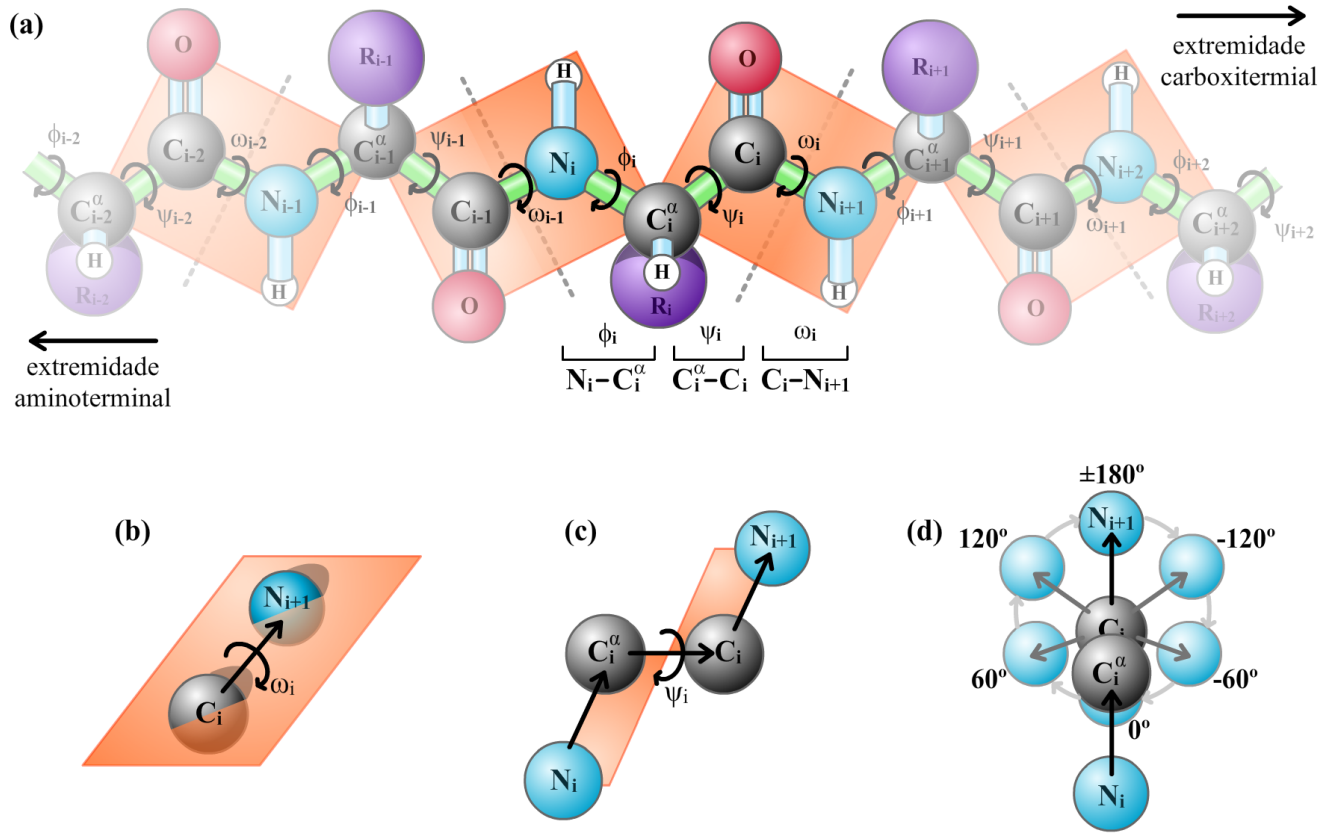
A estrutura das proteínas pode ser organizada de forma hierárquica, sendo que para cada nível da hierarquia, há um aumento da complexidade estrutural. Ainda, na Fórmula Estrutural 2.6, é apresentado novamente um esquema geral da estrutura molecular de uma proteína, com os átomos de carbono α destacados em azul. A cadeia linear dos componentes N_1-C_1^α-C_1-N_2-C_2^α-C_2-N_3-C_3^α-C_3-N_n-C_n^α-C_n forma o chamado esqueleto, cadeia principal, ou *backbone* da proteína e seus outros átomos formam a cadeia secundária ou cadeia lateral da proteína (LESK, 2016). Nota-se também que as ligações peptídicas estão desenhadas de forma pontilhada e em verde, representando, nesse caso, que essas ligações tem caráter de ligação dupla, implicando em não poder haver rotação entre os átomos envolvidos nesse eixo (NELSON; COX, 2014).



(2.6)

Além disso, a conformação da cadeia principal pode ser descrita inteiramente por ângulos de rotação entre as ligações $N_i - C_i^\alpha$, $C_i^\alpha - C_i$ e $C_i - N_{i+1}$ (ligação peptídica) de um resíduo i qualquer, sendo que esses ângulos diédricos de rotação são nomeados, respectivamente, de ângulos ϕ, ψ e ω. Na Figura 4 são apresentados quatro diagramas que representam tais ângulos.

Figura 4 – Ângulos ϕ, ψ e ω da cadeia principal de uma proteína: (a) Estrutura geral; (b) Grupo peptídico planar; (c) Ângulo diédrico ψ entre planos; e, (d) Limites de rotação do ângulo ψ

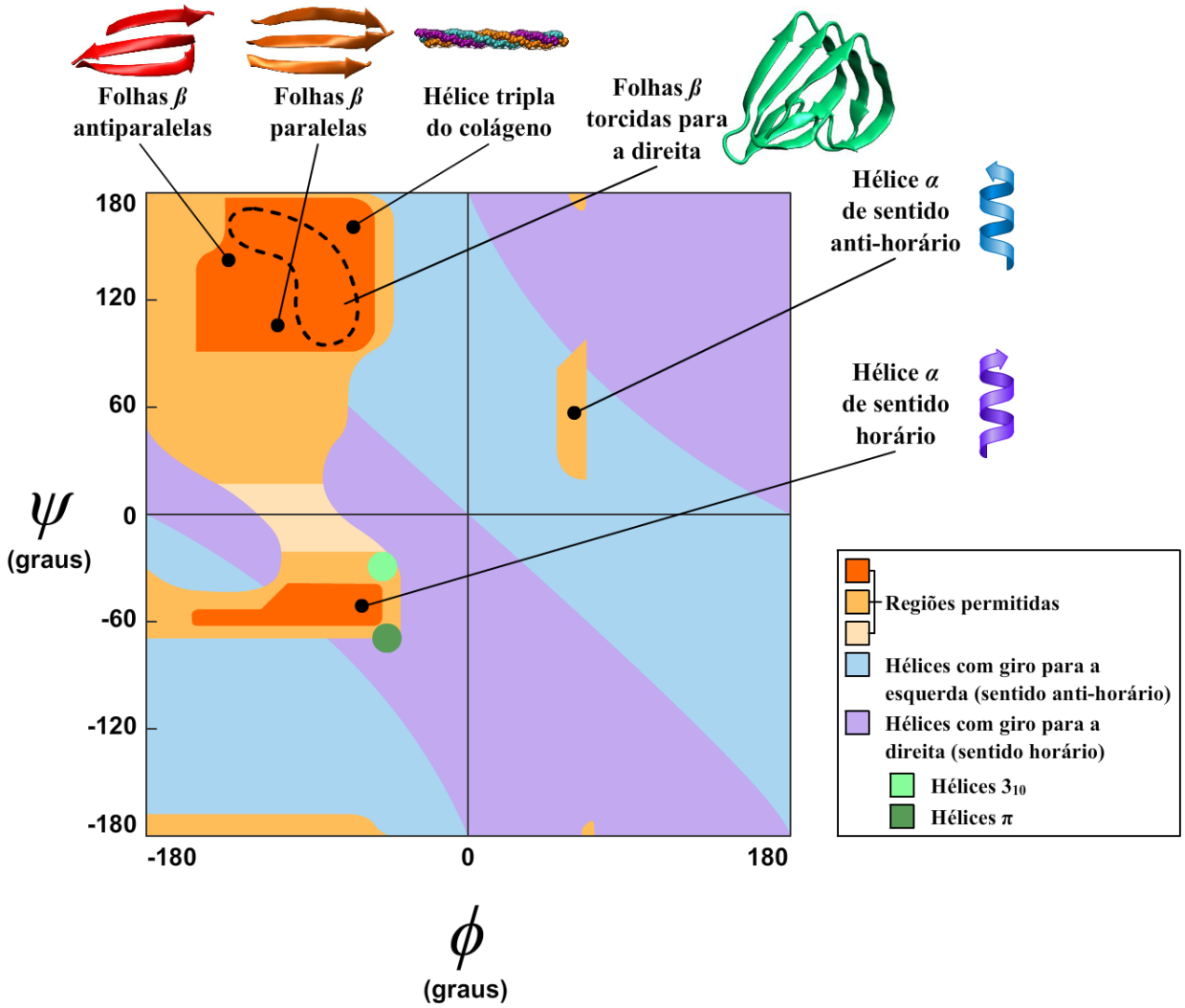


Fonte: Adaptada de Nelson e Cox (2014) pelo autor

Na Figura 4a é representado um recorte da estrutura de uma proteína hipotética. As ligações entre os átomos da cadeia principal estão coloridas em verde, enquanto as outras ligações, por exemplo, $C_i^\alpha - R_i$, estão coloridas em azul. Os ângulos de rotação ϕ, ψ e ω estão representados como setas circulares envolvendo suas respectivas ligações e os planos formados pelas ligações entre os átomos C_i e N_{i+1}, chamados de grupos peptídicos planares, estão representados em laranja. O detalhe de um destes planos é apresentado na Figura 4b, sendo que é importante frisar que a ligação do ângulo ω é sempre fixa em 180º ou 0º, não havendo rotação nesse eixo devido ao caráter de ligação dupla que existe na ligação peptídica. O detalhe do ângulo ψ entre os planos formados pelas ligações $N_i - C_i^\alpha$ e $C_i - N_{i+1}$ é mostrado na Figura 4c e, por fim, na Figura 4d, é mostrado como os ângulos ψ ou ϕ podem rotar, desde que não haja sobreposição espacial dos átomos.

Existem valores que são estericamente permitidos para os ângulos ϕ e ψ dos resíduos de uma proteína, sendo que as possibilidades de ângulos válidos para cada resíduo podem ser representadas pelo gráfico de Sasisekharan-Ramakrishnan-Ramachandran, normalmente denominado "Gráfico de Ramachandran" (LESK, 2016). Na Figura 5 é apresentado o esquema geral de um gráfico de Ramachandran.

Figura 5 – Gráfico de Ramachandran



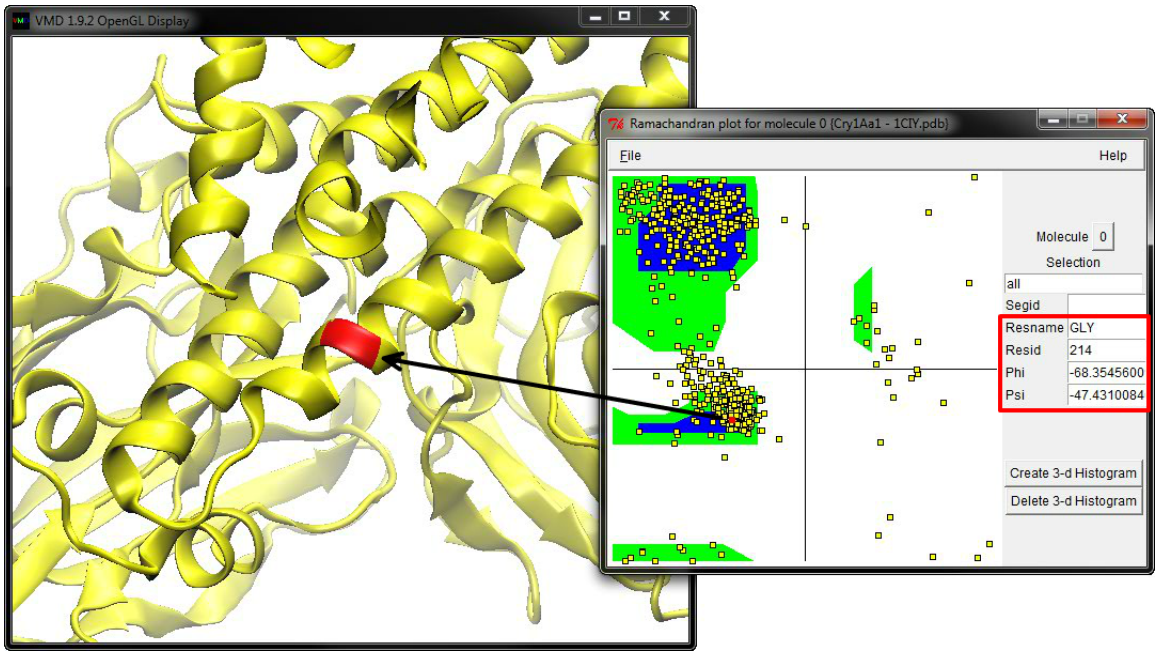
Fonte: Adaptada de Lesk (2016), Mathews et al. (2012) e Nelson e Cox (2014) pelo autor

No gráfico de Ramachandran apresentado na Figura 5, as regiões coloridas em laranja escuro representam conformações totalmente permitidas, pois não há sobreposição estérica entre átomos. As regiões coloridas em laranja médio indicam conformações que são permitidas, mas que são menos favoráveis que as conformações representadas pelas regiões laranja escuro. Ainda, a região colorida em laranja claro representa regiões ainda mais restritas. As regiões em azul e lilás representam regiões não permitidas, mas que se o fossem, abrigariam, respectivamente, estruturas helicoidais com torção à esquerda e com torção à direita. Como pode ser verificado no gráfico, os tipos de estruturas secundárias[5] que formam as proteínas aparecem em determinadas regiões, sendo que conformações do tipo β e da hélice tripla do colágeno estão situadas na região laranja escuro acima à esquerda (segundo quadrante), enquanto conformações do tipo α estão concentradas na região laranja escura inferior à esquerda (terceiro quadrante). Na Figura 6 pode ser visto

[5] Os tipos das estruturas secundárias que podem ocorrer nas proteínas serão apresentados no item "Estrutura Secundária" da lista de itens da página 45.

um exemplo de um gráfico de Ramachandran para a proteína Cry1Aa1 (PDB[6]: 1CIY).

Figura 6 – Exemplo de um gráfico de Ramachandran para o resíduo 214 (Gly) da proteína Cry1Aa1 (PDB: 1CIY)



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

À esquerda da Figura 6 encontra-se a representação gráfica da proteína, toda em amarelo, enquanto à direita, é apresentado o gráfico de Ramachandran correspondente a essa proteína. Pode-se perceber que o resíduo selecionado, representando por um quadrado em vermelho no gráfico, é uma glicina que está na posição 214 e têm ângulos $\phi = -68,3545600$ e $\psi = -47,4310084$. Uma seta indica, na estrutura da proteína, a localização deste resíduo, também destacado em vermelho. Todos os outros resíduos da proteína estão representados em amarelo, tanto na estrutura, quanto no gráfico. Em comparação ao modelo da Figura 5, é possível verificar que o resíduo selecionado está na região das conformações α, o que é confirmado no desenho da estrutura, mostrando o destaque em vermelho como componente de uma hélice α.

A seguir são apresentados os detalhes da hierarquia estrutural das proteínas, com base no que é apresentado nos trabalhos de Branden e Tooze (1999), Kessel e Ben-Tal (2011), Nelson e Cox (2014) e Lesk (2016):

- **Estrutura Primária:** A estrutura primária compreende a sequência de resíduos de aminoácidos, interligados por ligações peptídicas, que compõe a proteína, além

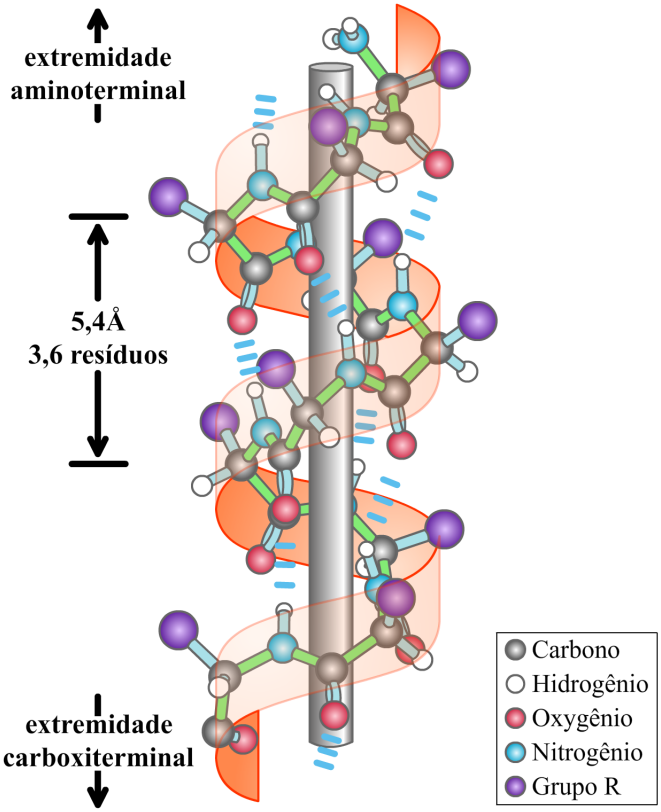
[6] Neste trabalho, serão apresentadas diversas representações gráficas de modelos tridimensionais de proteínas. Para cada uma dessas representações, será sempre citado o identificador *Protein Data Bank* (PDB) que é um código que identifica, de forma única, cada modelo proteico depositado no banco de dados de modelos de proteínas PDB, desenvolvido por Berman et al. (2000), e que está disponível no endereço <http://www.rcsb.org/> da Web.

de possíveis ligações dissulfeto que interligam aminoácidos de regiões diferentes da mesma sequência;

- **Estrutura Secundária:** A estrutura secundária de uma proteína é dada por quaisquer conjuntos de resíduos que compõe um segmento da estrutura primária. Esses segmentos apresentam uma organização espacial da cadeia principal da proteína, sem que se leve em consideração a interação de um segmento com qualquer outro, bem como em relação à localização das cadeias laterais dos resíduos. Os segmentos que constituem as estruturas secundárias assumem conformações comuns, o que implica em sequências de ângulos ϕ e ψ parecidos em cada tipo de organização, podendo ser classificados em vários tipos de subestruturas, sendo que as mais comuns são:

  - **Hélice $\boldsymbol{\alpha}$**: As hélices α são estruturas onde os resíduos estão organizados de forma helicoidal, circundando um eixo imaginário, tendo, para cada volta, cerca de 5,4Å de altura em relação a esse eixo e 3,6 resíduos. Para que isso aconteça a organização dos resíduos apresenta os valores aproximados de $\phi = -57^{o}$ e $\psi = -47^{o}$. Na Figura 7 pode ser vista uma representação gráfica de um modelo de uma hélice α.

Figura 7 – Modelo de uma hélice α



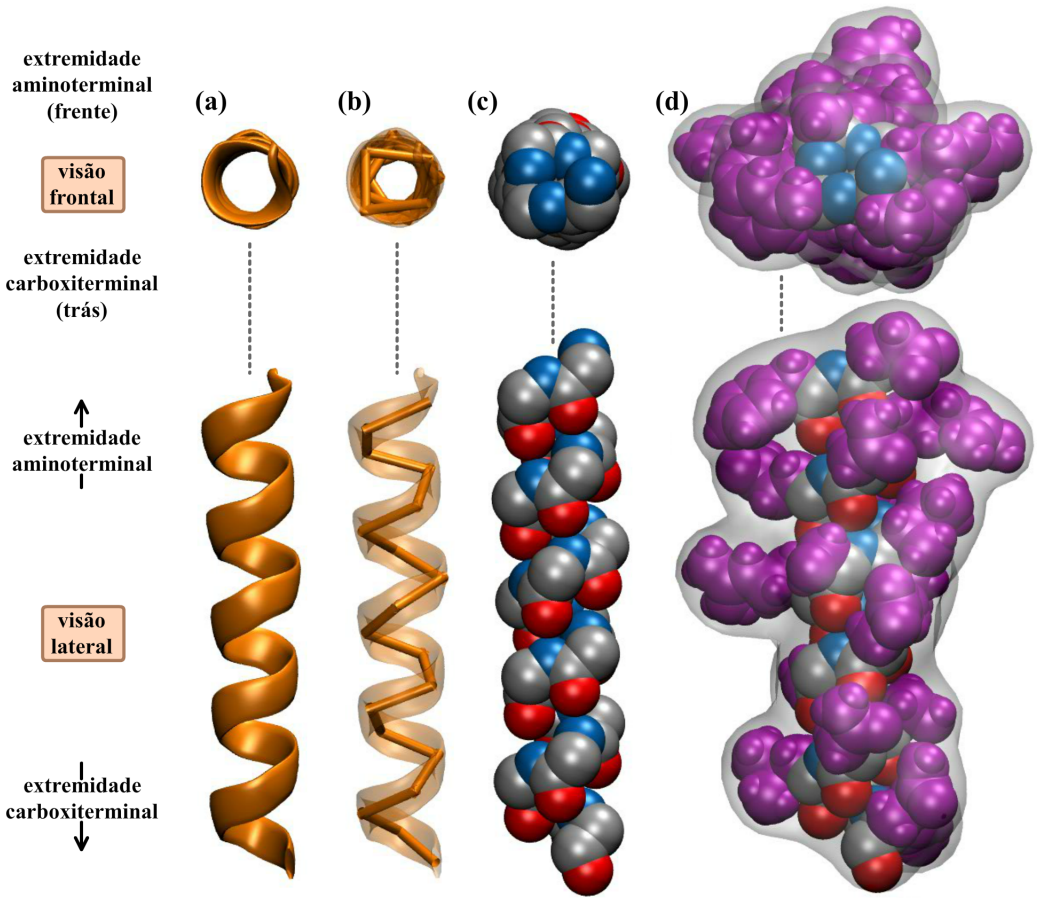
Fonte: Adaptada de Nelson e Cox (2014) pelo autor

A torção das hélices α, que sempre ocorre no sentido horário, é mantida por pontes de hidrogênio que são formadas entre o átomo de hidrogênio ligado ao

átomo de nitrogênio de um resíduo da cadeia principal (i), com o átomo de oxigênio do grupo carbonil do quarto aminoácido, a contar a partir do resíduo que contém o átomo de hidrogênio ($i+4$)[7], em direção à extremidade aminoterminal. Na Figura 8 podem ser vistas quatro representações diferentes, dividias em duas visões, de uma hélice α: na seção Figura 8a é utilizada a representação padrão, em forma de fita torcida, de uma hélice α. Essa representação é denominada do tipo "*cartoon*" nos diversos programas de visualização de estruturas de proteínas que existem. Na Figura 8b é utilizada a representação de traçagem ou de "*ribbon*" em que é possível verificar os eixos de curvatura da estrutura. Por fim, na Figura 8c e na Figura 8d, são feitas duas representações utilizando raios de van der Walls, sendo que na primeira, estão representados apenas os átomos da cadeia principal, enquanto na segunda, os átomos das cadeias laterais estão coloridos de roxo, além da superfície aproximada do segmento estar sendo apresentada;

[7] Deve-se considerar o resíduo i como o primeiro resíduo de uma volta.

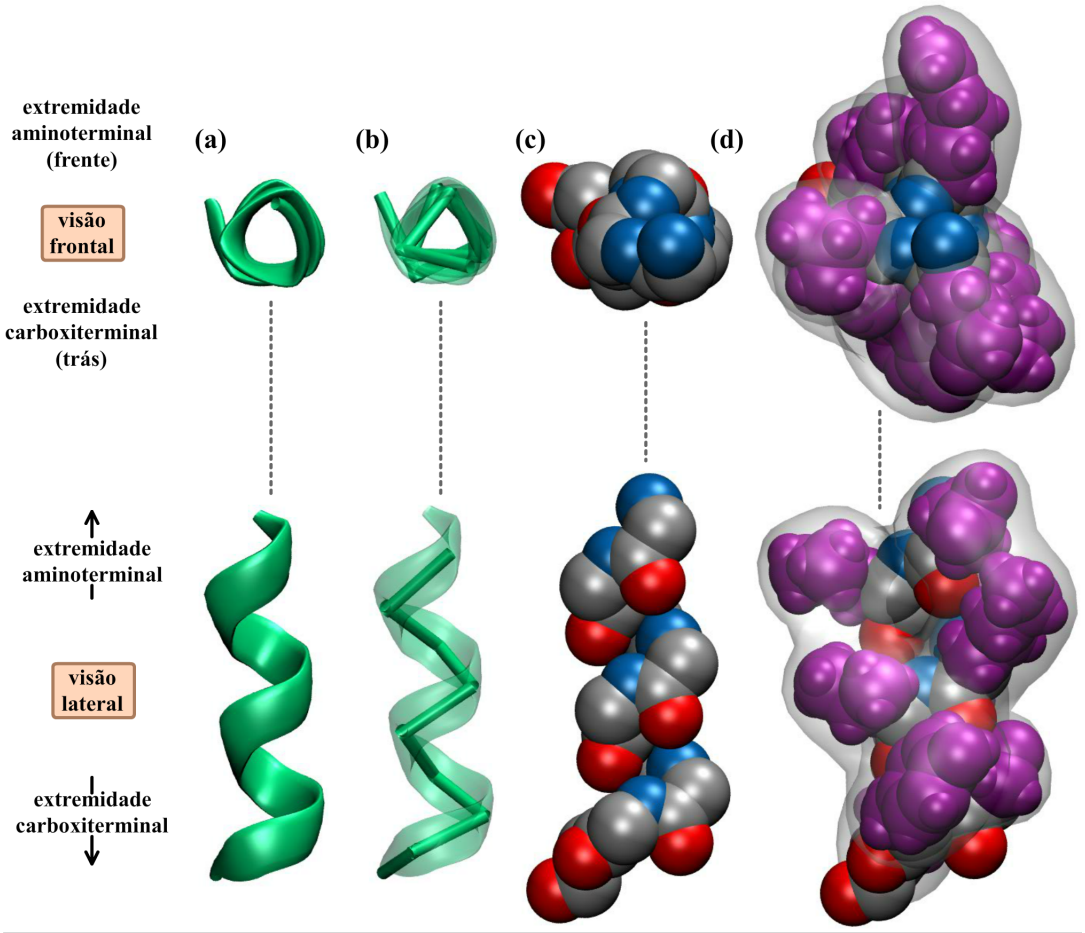
Figura 8 – Representações de uma hélice α compreendida entre os resíduos 90 e 108 da proteína Cry1Aa1 (PDB: 1CIY): (a) Visão em fitas; (b) Visão de traçagem; (c) Visão em raios de van der Walls da cadeia principal; e, (d) Visão completa em raios de van der Walls



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

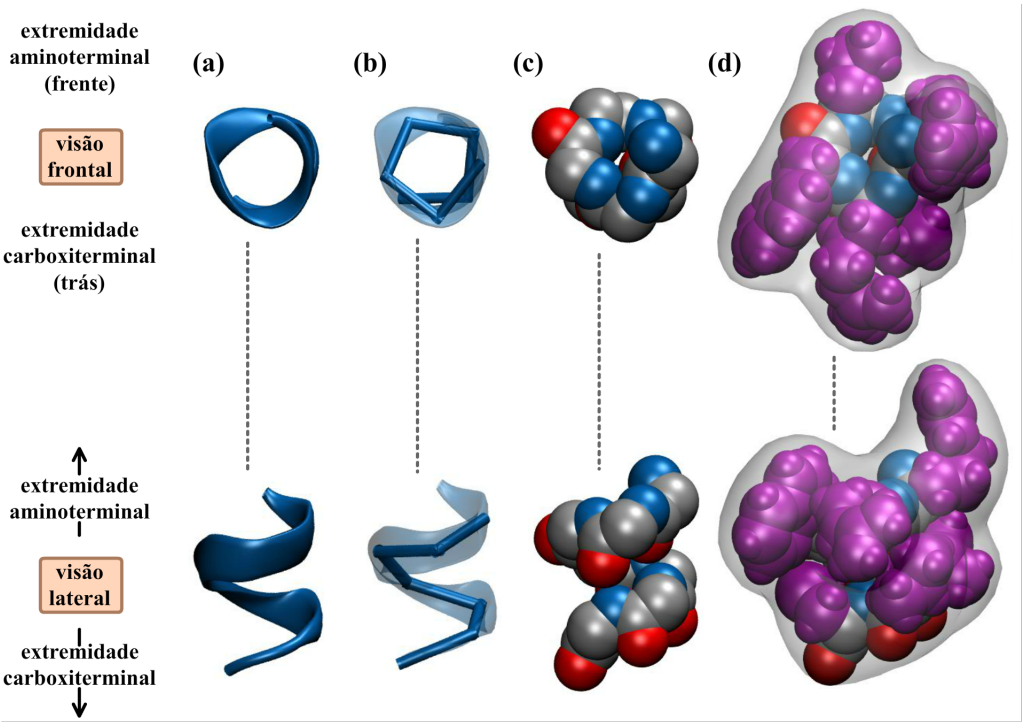
– **Outros tipos de hélices:** Além das hélices α, existem também outras conformações helicoidais, sendo que as hélices 3_{10} e π são dois exemplos relativamente comuns. Enquanto nas hélices α as pontes de hidrogênio distam em 4 resíduos $(i + 4)$ e cada volta tem aproximadamente 4 resíduos, nas hélices 3_{10} e π as pontes distam, respectivamente, em três $(i + 3)$ e cinco resíduos $(i + 5)$ e tem aproximadamente 3 e 5 resíduos por volta. Além disso, essas hélices são menos estáveis que as hélices α, visto que na hélice 3_{10} cada volta é mais "apertada", ou seja, a hélice é mais torcida, enquanto que, na hélice π as voltas são mais soltas, tornando a hélice mais frouxa. Essas características tornam as hélices 3_{10} e π menos estáveis, fazendo com que as mesmas sejam menos comuns que as hélices α, que no caso, possuem a configuração ótima em relação a estabilidade estrutural. Nas Figuras 9 e 10 esses dois tipos de hélices podem ser verificados, sendo que estão representados da mesma forma que a hélice α foi apresentada na Figura 8;

Figura 9 – Representações de uma hélice 3_{10} compreendida entre os resíduos 517 e 525 da proteína fosforilase b (PDB: 1ABB): (a) Visão em fitas; (b) Visão de traçagem; (c) Visão em raios de van der Walls da cadeia principal; e, (d) Visão completa em raios de van der Walls



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

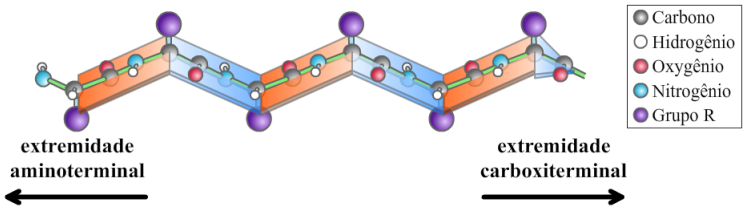
Figura 10 – Representações de uma hélice π compreendida entre os resíduos 489 e 495 da proteína fosforilase b (PDB: 1ABB): (a) Visão em fitas; (b) Visão de traçagem; (c) Visão em raios de van der Walls da cadeia principal; e, (d) Visão completa em raios de van der Walls



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

– **Conformação β**: Nas conformações β os resíduos de aminoácidos estão organizados em forma de zigue-zague, podendo se organizar lado a lado, formando assim as folhas β. Na Figura 11 pode ser visto um modelo de uma conformação β isolada, sendo que a ponta da seta do conjunto de planos imaginários da estrutura indica a direção para a extremidade carboxiterminal da cadeia da proteína;

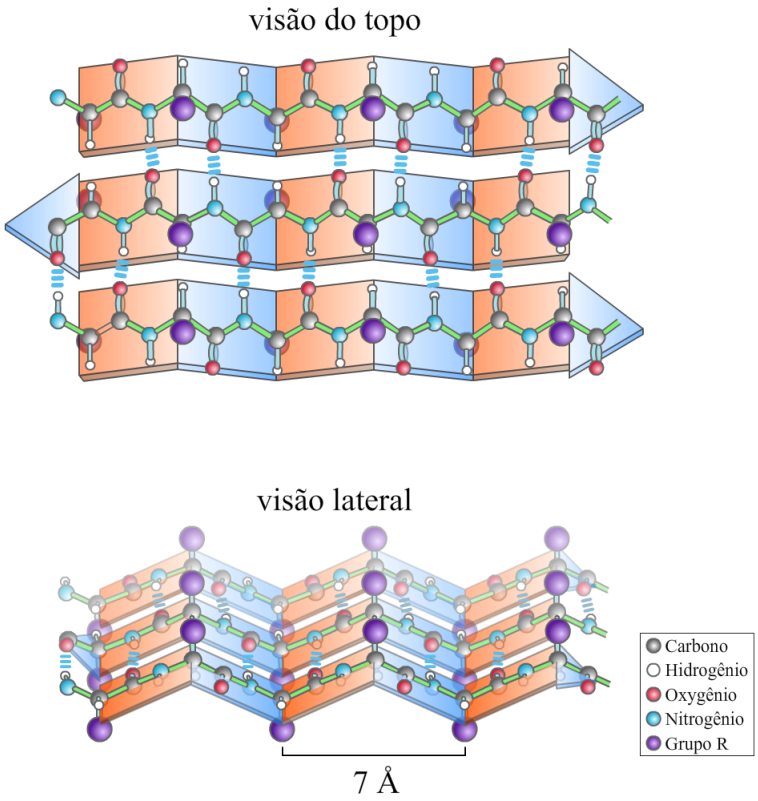
Figura 11 – Modelo de uma conformação β



Fonte: Adaptada de Nelson e Cox (2014) pelo autor

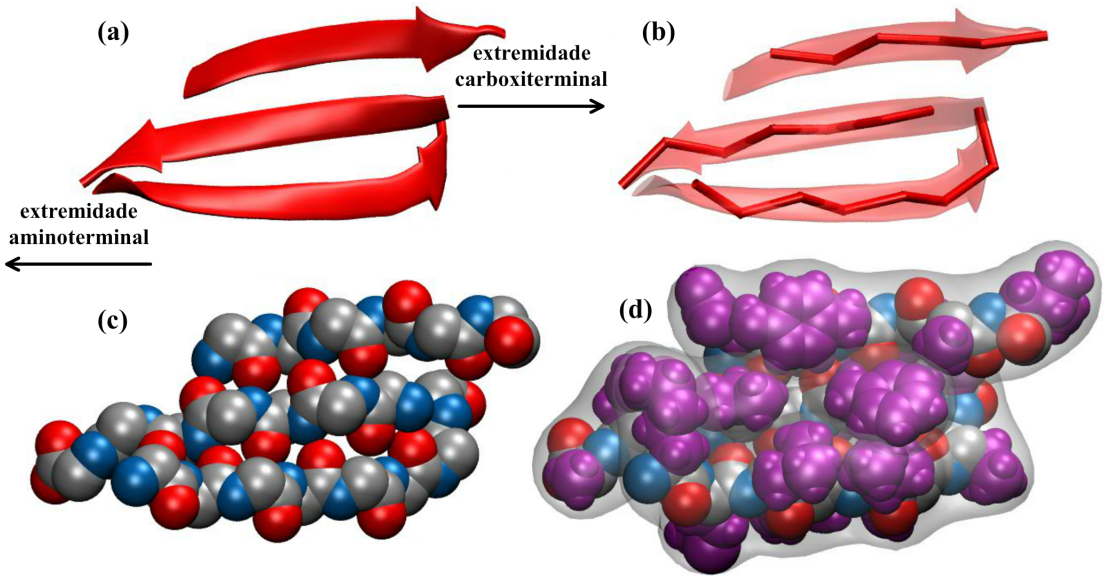
– **Folha β**: As folhas β são conjuntos de duas ou mais conformações β que estão ligadas lado a lado por pontes de hidrogênio. Normalmente, as folhas β são compostas por segmentos adjacentes da cadeia principal da proteína, entretanto, pode haver folhas β formadas por conformações β que, linearmente, estão distantes em relação à cadeia principal. A organização das conformações β na composição das folhas β pode se dar de duas formas: criando folhas β antiparalelas, quando, de forma intercalada, cada conformação β aponta para a direção contrária da anterior; e, quando a organização das conformações β se dá de forma paralela, ou seja, todas as conformações β constituintes da folha β apontam para a mesma direção, nesse caso, formando uma folha β paralela. Na Figura 12 pode-se verificar um modelo da constituição de uma folha β antiparalela, com visões a partir do topo e da perspectiva lateral, com destaque à distância de 7Å entre os grupos R dos resíduos espaçados dois a dois, e na Figura 13 pode-se ver uma folha β antiparalela representada em um *software* de visualização de estruturas de proteínas. Nessa figura, há quatro seções, apresentando uma visão a partir do topo desse tipo de folha β, sendo que na Figura 13a é apresentada a estrutura usando o desenho de fitas, na Figura 13b é apresentada a representação da traçagem dos trechos da cadeia principal, na Figura 13c e na Figura 13d é usada representação de raios de van der Walls, sendo que na primeira apenas as ligações da cadeia principal são mostradas e na segunda, além da superfície tridimensional da folha β, foram também inseridos os átomos das cadeias laterais, representados em roxo. Nas Figuras 14 e 15, de forma análoga, são apresentadas as mesmas representações para a folha β paralela;

Figura 12 – Modelo de uma folha β antiparalela



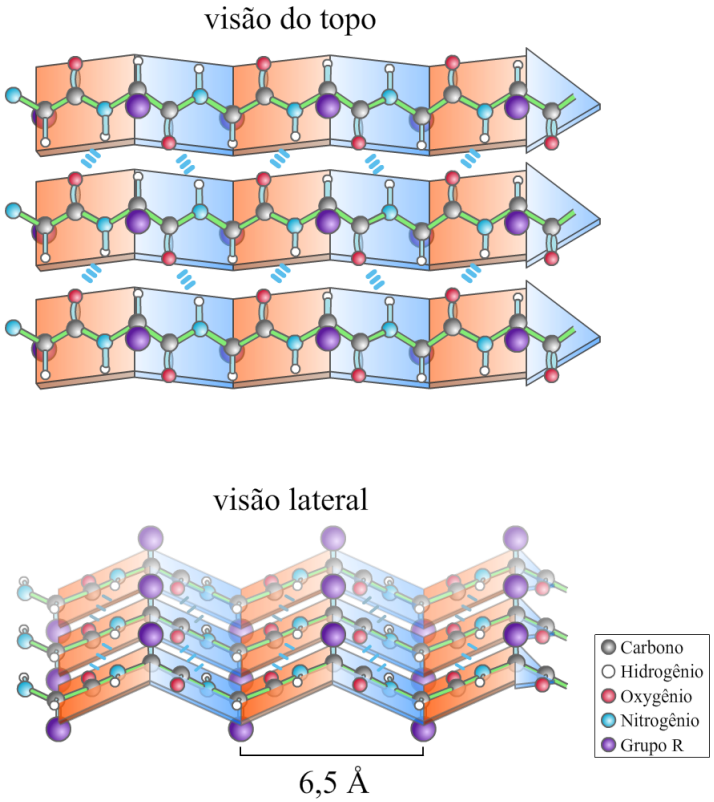
Fonte: Adaptada de Nelson e Cox (2014) pelo autor

Figura 13 – Representações de uma folha β antiparalela com três conformações β compreendidas entre os resíduos 167 e 174, 216 e 221 e 286 e 300 da proteína histamina-metiltransferase humana (PDB: 1JQD): (a) Visão em fitas; (b) Visão de traçagem; (c) Visão em raios de van der Walls da cadeia principal; e, (d) Visão completa em raios de van der Walls



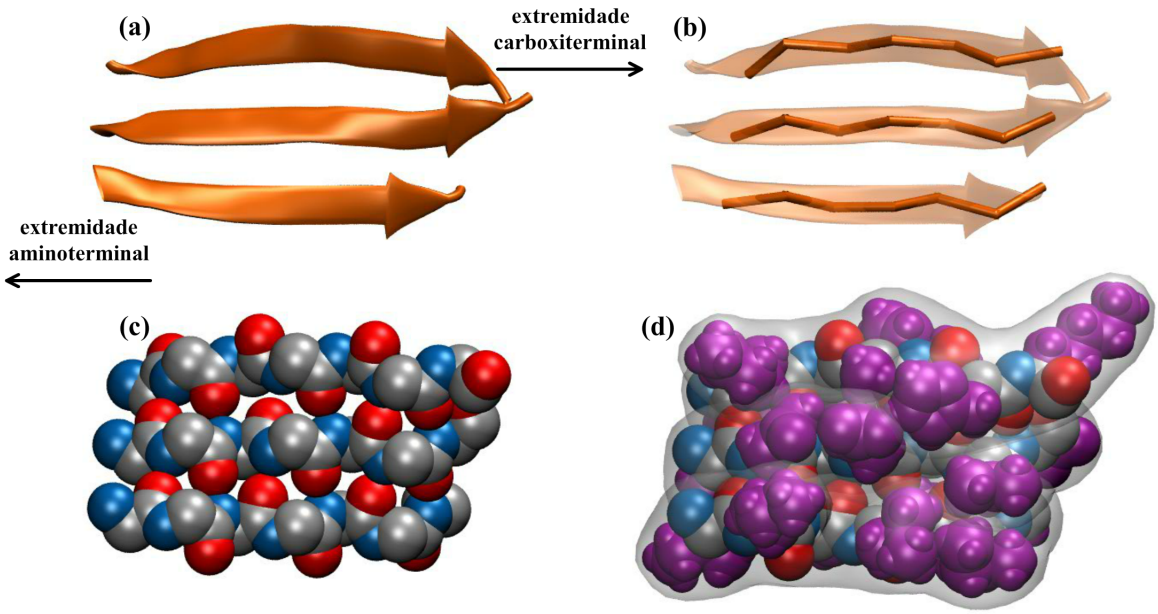
Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

Figura 14 – Modelo de uma folha β paralela



Fonte: Adaptada de Nelson e Cox (2014) pelo autor

Figura 15 – Representações de uma folha β paralela com três conformações β compreendidas entre os resíduos 54 e 60, 83 e 90 e 111 e 118 da proteína histamina-metiltransferase humana (PDB: 1JQD): (a) Visão em fitas; (b) Visão de traçagem; (c) Visão em raios de van der Walls da cadeia principal; e, (d) Visão completa em raios de van der Walls

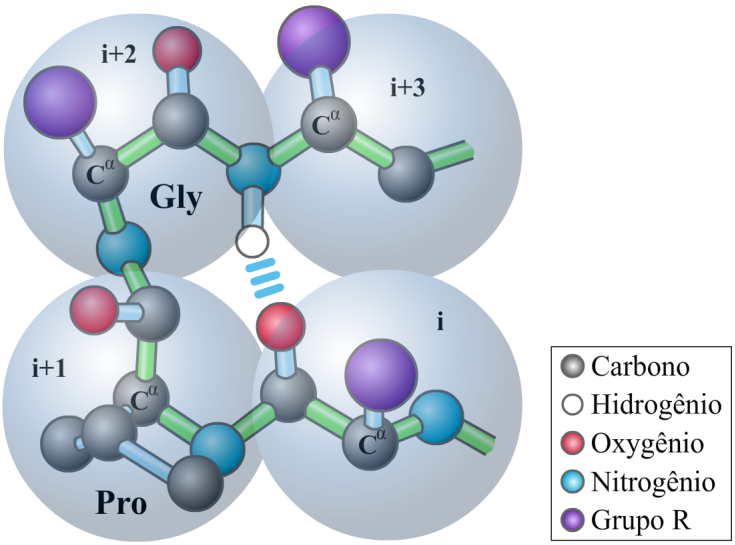


Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

– **Volta $\boldsymbol{\beta}$**: Por fim, outra estrutura secundária recorrente que está presente na estrutura das proteínas é a volta β, composta por quatro resíduos, e que é responsável pelas ligações entre as outras subestruturas constituintes da proteína. Na Figura 16 pode ser visto um modelo dessa volta, sendo que, normalmente, o segundo resíduo $(i+1)$ é uma prolina e o terceiro $(i+2)$ é uma glicina além de a ponte de hidrogênio ser formada entre os resíduos 1 (i) e 4 $(i+3)$. Existem ainda as voltas α, γ, δ e π, menos comuns que as voltas β, e que tem como características: volta α, cinco resíduos com ponte de hidrogênio entre os resíduos 1 (i) e 5 $(i+4)$; volta γ, três resíduos com ponte de hidrogênio entre os resíduos 1 (i) e 3 $(i+2)$; volta δ[8], dois resíduos com ponte de hidrogênio entre os resíduos 1 (i) e 2 $(i+1)$; e, volta π, seis resíduos com ponte de hidrogênio entre os resíduos 1 (i) e 6 $(i+5)$. Na Figura 17a é apresentada uma volta β, em azul claro, ligando uma extremidade de uma hélice α, em roxo, a uma extremidade de uma conformação β, em amarelo. Na Figura 17b são apresentadas os resíduos dessa volta β, com destaque à prolina, em verde, e à glicina, em azul. Por fim, na Figura 17c é mostrada novamente a volta β, agora com toda a sua estrutura e superfície;

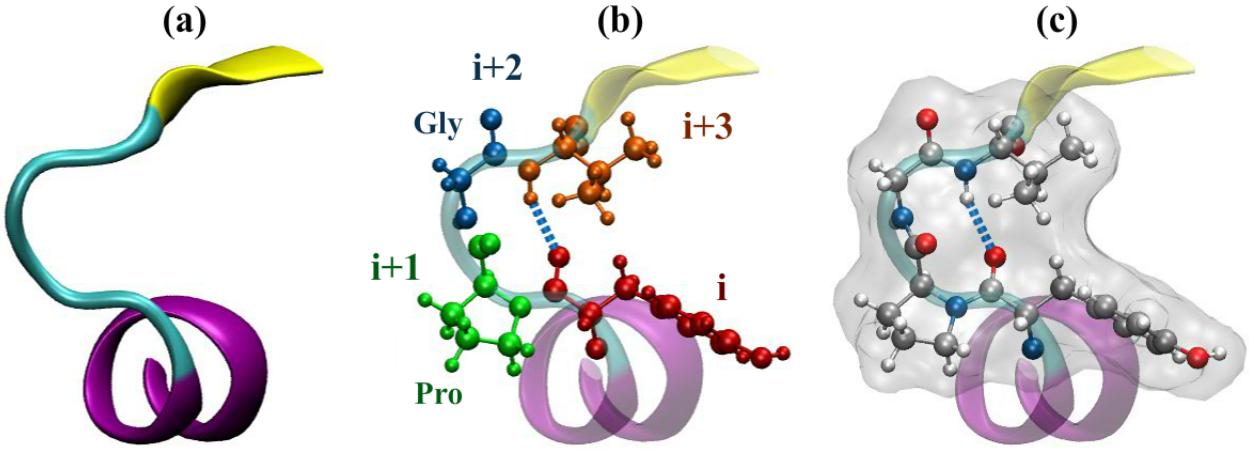
[8] É uma estrutura teórica, visto que não é estericamente permitida.

Figura 16 – Modelo de uma volta β



Fonte: Adaptada de Nelson e Cox (2014) pelo autor

Figura 17 – Representações de uma volta β compreendida entre os resíduos 78 e 81 da proteína histamina-metiltransferase humana (PDB: 1JQD): (a) Visão em fitas; (b) Visão estrutural com resíduos destacados; e, (c) Visão estrutural completa



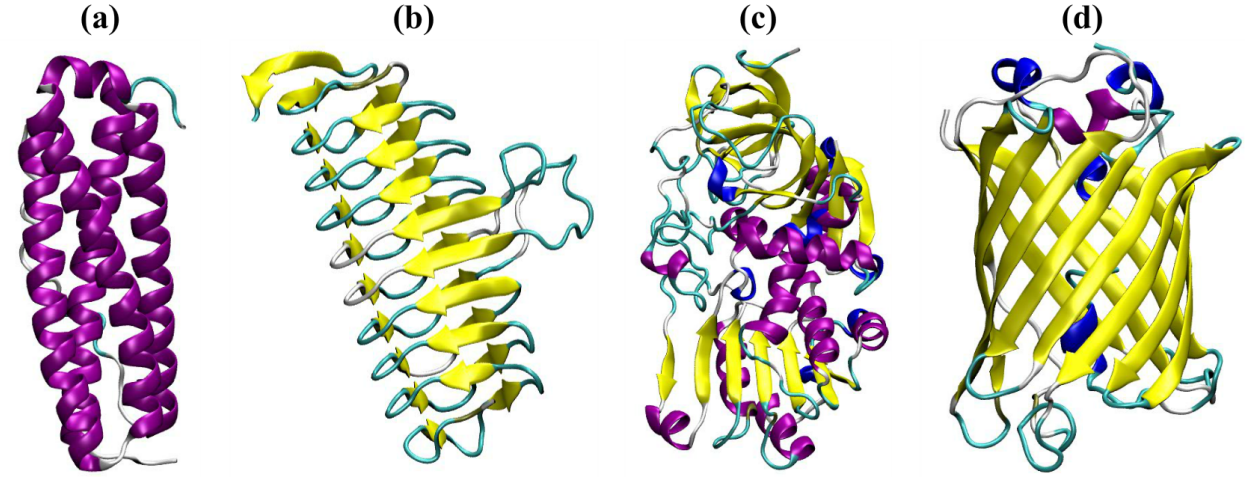
Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

– **Estruturas Supersecundárias:** As estruturas supersecundárias, ou também denominadas de motivos[9], são agrupamentos de conjuntos de estruturas secundárias que fazem parte da estrutura completa de uma proteína e que aparecem de forma recorrente em diversas proteínas. Eles são classificados de acordo com os tipos de estruturas secundárias que os compõe, podendo ser: todo α, quando a maior parte do motivo é composto apenas por conformações α (hélices); todo β, quando a maior parte do motivo é composto por conformações β; α/β, composto de conformações α e β, organizadas de forma intercalada, além de que

[9] Tradução do termo *motif* da língua inglesa.

as conformações β estão organizadas, na maioria das vezes, na forma de folhas β paralelas; e, $\alpha + \beta$, em que conjuntos de conformações α e β se intercalam, sendo que nessa classificação as conformações β estão organizadas na forma de folhas β antiparalelas. Além disso, os motivos são utilizados para classificar as estruturas das proteínas de acordo com o banco de dados de classificação estrutural *Structural Classification of Proteins* (SCOP)[10]. Essas classes de motivos podem ser verificadas de forma gráfica, respectivamente, na Figura 18a, na Figura 18b, na Figura 18c e na Figura 18d;

Figura 18 – Representações das quatro classes de motivos: (a) Todo α, cadeia A da proteína Bacterioferritina (PDB: 1BCF); (b) Todo β, resíduos 1 a 198 da proteína UDP *N*-acetilglucosamina-aciltransferase (PDB: 1LXA); (c) α/β, cadeia A da proteína Álcool-desidrogenase (PDB: 1DEH); e, (d) $\alpha + \beta$, proteína verde fluorescente de água-viva (*Aequorea victoria*) (PDB: 1EMA)

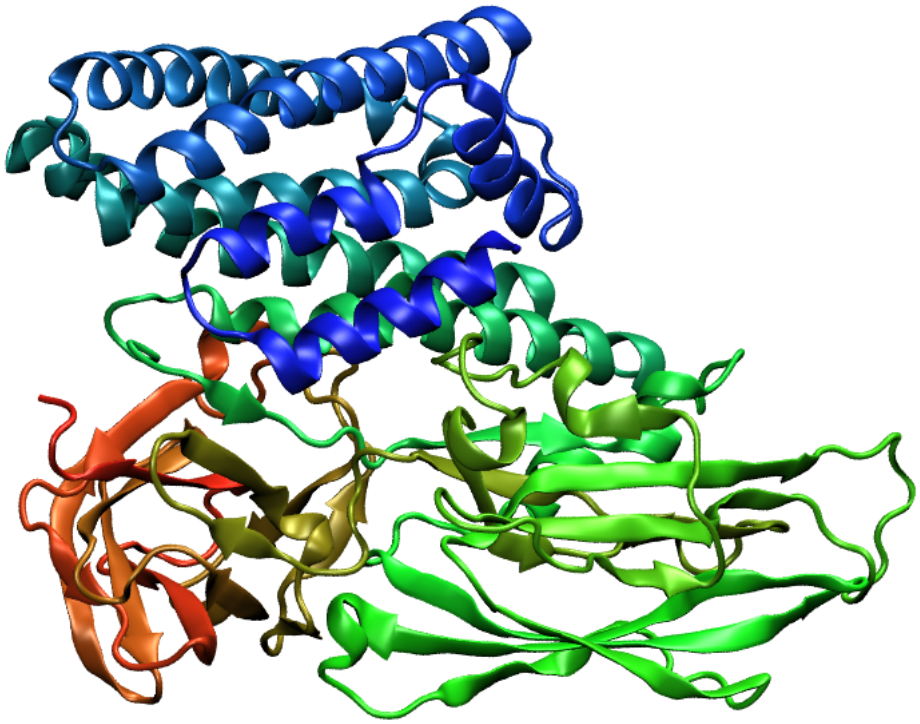


Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

- **Estrutura Terciária:** A estrutura terciária de uma proteína é a organização completa da estrutura tridimensional da mesma em seu estado nativo, indo na extremidade n-terminal à extremidade c-terminal. Na Figura 19 pode ser vista a representação gráfica da estrutura completa da proteína Cry1Aa1, colorida de forma a mostrar as regiões da estrutura de acordo com o acompanhamento da cadeia principal, visto que a extremidade n-terminal está colorida em azul, dirigindo-se ao meio da proteína em verde até chegar na extremidade c-terminal, em vermelho;

[10] O banco de dados SCOP foi desenvolvido por Murzin et al. (1995) e pode ser acessado no endereço <http://scop.mrc-lmb.cam.ac.uk/scop/> da Web. Outro banco de dados de classificação estrutural importante é o *Class Architecture Topology Homologous superfamily* (CATH) desenvolvido por Sillitoe et al. (2015) e que é acessado a partir do endereço <http://www.cathdb.info/>.

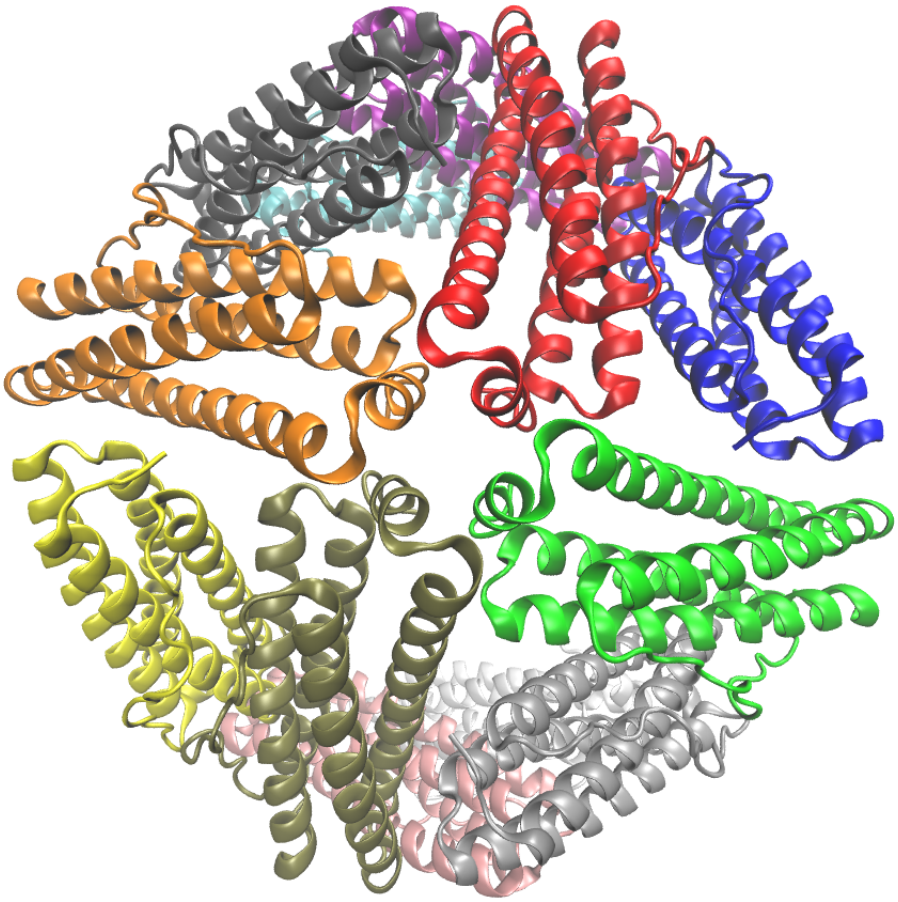
Figura 19 – Estrutura completa da proteína Cry1Aa1 com extremidade n-terminal em azul e extremidade c-terminal em vermelho (PDB: 1CIY)



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

- **Estrutura Quaternária:** Algumas proteínas possuem diversas subunidades terciárias, chamadas de cadeias, e por isso estão organizadas em uma estrutura quaternária, a qual engloba todas as estruturas terciárias que as compõe. Quando uma proteína possui mais de uma unidade terciária, ela recebe o nome de multímero. Em relação às cadeias formadoras da estrutura quaternária, elas são nomeadas utilizando letras maiúsculas do alfabeto latino. Por exemplo, na Figura 20, é apresentada a proteína Bacterioferritina de *Escherichia coli* que possui doze cadeias, cada uma colorida de uma cor distinta, sendo nomeadas de cadeia A até cadeia L.

Figura 20 – Estrutura quaternária da proteína Bacterioferritina com cada uma das doze cadeias coloridas com uma cor distinta (PDB: 1BCF)



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

Além dessa classificação, ainda existe o conceito de domínio que, segundo (NELSON; COX, 2014), foi cunhado por Jane Richardson no ano de 1981, sendo definido como uma parte de um polipeptídeo que é estável de forma independente, além de poder ter certo grau de liberdade de movimento em relação às outras partes da estrutura proteica. Normalmente as proteínas possuem dois ou mais domínios em suas estruturas, sendo que cada domínio apresenta, na maioria das vezes, funções distintas e, caso a estrutura da proteína seja clivada de modo a separar a região de um domínio, eles têm a característica de manter suas conformações independentemente do restante da cadeia polipeptídica. Há também proteínas com apenas um domínio por causa do tamanho de suas cadeias, sendo que o domínio, nesses casos, é a própria proteína. Além disso, do ponto de vista conformacional, os domínios podem aparecer como aglomerados distintos na proteína, mas também há a possibilidade de haver uma grande área de contato entre dois ou mais domínios, sendo difícil de, a partir somente da visualização da estrutura tridimensional da proteína, distinguir os domínios que a compõe. Por fim, ainda é possível verificar na literatura os termos *coil* e *loop*. Esses termos são usados para indicar regiões degeneradas que não tem uma

estrutura secundária bem definida, podendo ser composto por um conjunto de voltas, e que aparecem como regiões de ligação entre componentes da estrutura secundária que são bem definidos (hélices α e conformações β).

Vale também destacar que a determinação das estruturas das proteínas pode ser feita utilizando métodos experimentais como a Difração em Raios X ou a Ressonância Magnética Nuclear (RMN) (NELSON; COX, 2014; LESK, 2016), gerando modelos próximos ou totalmente iguais às estruturas nativas das proteínas, obtendo assim resultados mais acurados do que métodos *in silico* para predição estrutural de proteínas, a partir da estrutura primárias das mesmas, que, apesar de úteis para se ter uma ideia de como é a conformação de uma proteína, acabam não sendo precisos o bastante para representar de forma fidedigna a estrutura de uma proteína que precisa ser estudada de forma mais detalhada.

Pelo exposto, pode-se verificar que a organização estrutural das proteínas confere as características conformacionais das mesmas, influenciando assim, diretamente, na função de cada uma. Nota-se então que a possibilidade de se comparar as estruturas de diferentes proteínas deve ser algo importante, para, por exemplo, explicar o motivo de duas proteínas, que têm estruturas primárias diferentes, terem funções iguais ou parecidas, ou quais as diferenças estruturais fazem com que duas proteínas homólogas atuem com a mesma função, mas de forma específica, em diferentes organismos. Antes de se explorar a importância da comparação estrutural de proteínas e como essas comparações podem ser feitas, faz-se necessário apresentar a ideia de comparação/alinhamento de sequências (estruturas primárias) de diferentes proteínas.

2.1.2 Comparação ou Alinhamento de Sequências

O alinhamento de sequências tem o objetivo de identificar regiões similares de DNAs, RNAs ou proteínas, permitindo assim identificar relações funcionais, estruturais e evolucionárias entre elas. Para realizar o alinhamento, ou a comparação, das estruturas primárias das proteínas, são usados os algoritmos[11] de alinhamento de sequências, que permitem que duas sequências distintas, uma de cada proteína, sejam comparadas, alinhando seções comuns entre as duas, criando assim um padrão que pode ser analisado, permitindo verificar algum relacionamento entre as proteínas comparadas (LESK, 2016). Esses algoritmos conseguem alcançar esse objetivo ao inserir espaços (*gaps*[12]) nas sequências de forma a permitir um alinhamento de mesmo comprimento, o que facilita a comparação das sequências (LESK, 2016).

[11] Sequência de passos ou instruções, organizados de forma lógica, que ao serem executados conseguem resolver um problema.

[12] Os *gaps* (lacunas) são espaços inseridos nos processos de alinhamento de proteínas, tanto sequenciais quanto estruturais, para que se consiga alinhar regiões similares nas duas proteínas que não estão localizadas nas mesmas posições relativas das sequências/estruturas.

Para que a implementação computacional desses algoritmos seja viabilizada, existe a necessidade de se utilizar uma técnica de programação chamada Programação Dinâmica (PD), que consiste na reutilização de dados preprocessados para o obtenção de novos dados. Por exemplo, para uma proteína A com sequência $A_1 \dots A_p$ e uma proteína B com sequência $B_1 \dots B_q$, define-se uma matriz M_{pq}, onde cada valor M_{ij} contém o escore da comparação entre A_i e B_j (RUSSELL; BARTON, 1992).

Existem duas abordagens básicas para o alinhamento/comparação de sequências de proteínas, sendo que as mesmas serão apresentadas a seguir.

2.1.2.1 Alinhamento Global

O alinhamento global tem o objetivo de verificar toda a extensão das sequências a serem comparadas, alinhando-as por completo (NEEDLEMAN; WUNSCH, 1970). Nesta abordagem, o objetivo é alinhar toda a sequência s^1 sobre a sequência s^2, de tal forma que seja possível aplicar uma avaliação do grau de similaridade entre elas. Em 1970, Needleman e Wunsch propuseram um algoritmo baseado em PD que busca encontrar uma solução ótima para o problema. A técnica é apresentada em Needleman e Wunsch (1970) e o texto a seguir é baseado no mesmo trabalho, sendo que as equações apresentadas foram usadas como exemplo para que o algoritmo seja exemplificado. Na Equação 2.1 é apresentada uma função recursiva que é utilizada para esse cálculo.

$$\begin{aligned} F(0,0) &= 0 \\ F(i,j) &= max \begin{cases} F(i-1,j-1) + s(s_j^1, s_i^2) \\ F(i-1,j) + d \\ F(i,j-1) + d \end{cases} \end{aligned} \tag{2.1}$$

Onde:

- $\boldsymbol{F(i-1,j-1) + s(s_i^1, s_j^2)}$**:** quando s_i^1 alinha com s_j^2, somando o escore obtido entre s_i^1 e s_j^2;
- $\boldsymbol{F(i-1,j) + d}$**:** quando s_i^1 alinha com um *gap*;
- $\boldsymbol{F(i,j-1) + d}$**:** quando s_j^2 alinha com um *gap*.

A seguir é apresentada uma simulação da aplicação dessa função recursiva utilizando a PD. Deve-se considerar as seguintes sequências e parâmetros para realizar a simulação:

- $s^1 =$ AGC[13];

[13] Os algoritmos de alinhamento serão exemplificados utilizando sequências de nucleotídeos ao invés resíduos de aminoácidos, permitindo que a matriz de substituição seja apresentada de forma mais sucinta, pois tem dimensões 4×4, ao invés de dimensões 20×20 caso fosse utilizada uma matriz de substituição para aminoácidos em um exemplo envolvendo proteínas.

- $s^2 = \text{AAG}$;
- $d = -5$ (*gap penalty*);
- $s(s_i^1, s_j^2) = m$ é a função de escore, onde m é um valor obtido cruzando os valores de s_i^1 e s_j^2 em uma matriz de substituição.

Para a simulação utiliza-se uma tabela, entretanto é importante reforçar que o algoritmo faz uso de uma matriz, chamada de matriz de PD, além de recursão para o processamento das sequências. Na Figura 21 pode ser visto o início da simulação.

Figura 21 – (a) Matriz de PD e (b) matriz de substituição para a simulação do algoritmo de Needleman-Wunsch – Etapa 1

		A	A	G
	0	-5	-10	-15
A	-5			
G	-10			
C	-15			

(a)

Matriz de Substituição				
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

(b)

Fonte: Elaborada pelo autor

A primeira etapa da execução do algoritmo inicializa a matriz de PD. Na Figura 21 é apresentada a simulação do algoritmo, sendo que na Figura 21a, é representada a inicialização da matriz e na Figura 21b, e nas figuras da simulação subsequentes, é mostrada a tabela de escore entre as bases nitrogenadas A, T, C e G do DNA. De acordo com a Equação 2.1, $F(0,0) = 0$, sendo assim, a célula da linha 0 e da coluna 0 da matriz deve ser inicializada com o valor zero. O restante da tabela foi calculado considerando o valor de *gap penalty*, definido em -5, e aplicado a cada sequência que alinha com um *gap*. É possível observar que o *gap penalty* é acumulativo, ou seja, para a célula $(1,0)$, têm-se $(1,0) = (0,0)+(-5)$, para a célula $(2,0)$, têm-se $(2,0) = (1,0)+(-5) = (-5)+(-5) = -10$, e assim sucessivamente.

Na Figura 22a é apresentado o cálculo do algoritmo considerando os valores de i e j, valendo respectivamente 1 e 1.

Figura 22 – (a) Matriz de PD e (b) matriz de substituição para a simulação do algoritmo de Needleman-Wunsch – Etapa 2

		A	A	G
	0	-5	-10	-15
A	-5	$F(1,1)$		
G	-10			
C	-15			

(a)

Matriz de Substituição				
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

(b)

Fonte: Elaborada pelo autor

Nesse cenário, a função recursiva é aplicada e o valor máximo obtido é 2, conforme ilustrado no cálculo realizado na Equação 2.2.

$$F(1,1) = max \begin{cases} F(1-1,1-1) + s(s_1^1, s_1^2) = F(0,0) + 2 = 0 + 2 = 2 \\ F(1-1,1) + d = F(0,1) + (-5) = -5 - 5 = -10 \\ F(1,1-1) + d = F(1,0) + (-5) = -5 - 5 = -10 \end{cases} = 2 \qquad (2.2)$$

É importante verificar qual caminho recursivo obteve o maior escore e anotar na tabela usando uma seta para identificar de qual célula o valor foi originado. Analisando as três possibilidades da função recursiva apresentada na Equação 2.1, tem-se os caminhos possíveis que são apresentados na Figura 23.

Figura 23 – Precedentes de acordo com o caminho recursivo

$$\begin{array}{cc} F(i-1,j-1) + s(s_i^1, s_j^2) & \searrow \\ F(i-1,j) + d & \downarrow \\ F(i,j-1) + d & \rightarrow \end{array}$$

Fonte: Elaborada pelo autor

Nesse exemplo, a chamada $F(i-1,j-1) + s(s_i^1, s_j^2))$, obteve o maior escore (valor 2), por isso a seta é posicionada na célula $(0,0)$ da matriz, ou seja, a célula esquerda superior, indicando que o valor da célula $(1,1)$ foi calculado a partir da célula $(0,0)$. O valor e a seta são apresentados respectivamente na Figura 24a e na Figura 24b que é a matriz de precedência.

Figura 24 – (a) Matriz de PD, (b) matriz de precedência e (c) matriz de substituição para a simulação do algoritmo de Needleman-Wunsch – Etapa 3

		A	A	G
	0	-5	-10	-15
A	-5	2		
G	-10			
C	-15			

(a)

		A	A	G
A		↘		
G				
C				

(b)

Matriz de Substituição				
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

(c)

Fonte: Elaborada pelo autor

Dando continuidade à simulação, na Figura 25 é apresentado o cálculo referente a célula $(1, 2)$.

Figura 25 – (a) Matriz de PD, (b) matriz de precedência e (c) matriz de substituição para a simulação do algoritmo de Needleman-Wunsch – Etapa 4

		A	A	G
	0	-5	-10	-15
A	-5	2	-3	
G	-10			
C	-15			

(a)

		A	C	G
A		↘	↘, →	
C				
G				

(b)

Matriz de Substituição				
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

(c)

Fonte: Elaborada pelo autor

Observa-se que na Figura 25b, na posição (1,2), existem duas setas precedentes alcançando as células $(2, 3)$ e $(1, 3)$. Na Equação 2.3 são apresentadas as funções recursivas que tiveram o maior escore para o cálculo dessa célula. É possível observar que duas chamadas de função distintas, $F(i-1, j-1) + s(s_i^1, s_j^2)$ e $F(i, j-1) + d$, obtiveram o

mesmo escore, no caso -3.

$$F(1,2) = max \begin{cases} F(1-1,2-1) + s(s1_1, s2_2) = F(0,1) + 2 = -5 + 2 = -3 \\ F(1-1,2) + d = F(0,2) + (-5) = -10 - 5 = -15 \\ F(1,2-1) + d = F(1,1) + (-5) = 2 - 5 = -3 \end{cases} = -3 \tag{2.3}$$

Por esse motivo, a célula $(1,2)$ possui duas setas precedentes chegando à mesma. Em termos de alinhamento, isso significa que a partir da célula $(1,2)$ da matriz de PD é possível alinhar a sequência de duas maneiras distintas.

Por fim, na Figura 26 é apresentado o resultado final do algoritmo.

Figura 26 – (a) Matriz de PD, (b) matriz de precedência e (c) matriz de substituição para a simulação do algoritmo de Needleman-Wunsch – Etapa Final

		A	A	G
	0	-5	-10	-15
A	-5	2	-3	-8
G	-10	-3	-3	-1
C	-15	-8	-8	-6

(a)

		A	A	G
A		↘	↘, →	→
G		↓	↘	↘
C		↓	↓	↓

(b)

Matriz de Substituição				
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

(c)

Fonte: Elaborada pelo autor

O alinhamento ótimo deve ser processado a partir da última célula da matriz, nesse caso, a célula $(3,3)$. O processo para encontrar o melhor ou os melhores alinhamentos ocorre recursivamente a partir da última célula da matriz terminando quando o algoritmo alcança a posição (0,0). As regras para alinhamento são definidas da seguinte forma:

a) **Seta diagonal**: se s_i^1 alinha com s_j^2;

b) **Seta vertical**: se s_i^1 alinha com espaço (*gap*);

c) **Seta horizontal**: se s_j^2 alinha com espaço (*gap*).

Considerando isso, é possível identificar dois alinhamentos na simulação executada. Os alinhamentos possíveis são apresentados na Figura 27.

Figura 27 – Alinhamentos obtidos após a execução do algoritmo de Needleman-Wunsch

Fonte: Elaborada pelo autor

Ambos os alinhamentos são equivalentes, ou seja, possuem o mesmo escore, nesse caso -6. O valor do escore pode ser obtido inspecionando a célula da última linha e da última coluna da matriz de PD, nesse caso, a célula $(3, 3)$.

2.1.2.2 Alinhamento Local

Os alinhamentos locais, por sua vez, buscam realizar o alinhamento de um ou mais segmentos das sequências, realizando uma busca por regiões similares e apresentando aquelas que forem mais parecidas (SMITH; WATERMAN, 1981), diferente do alinhamento global que busca alinhar as sequências do início ao fim. O algoritmo Smith-Waterman, desenvolvido em 1981, é uma evolução do algoritmo de Needleman-Wunsch e foi projetado para permitir alinhamento local entre sequências. A seguir será apresentado este algoritmo, descrito no trabalho de (SMITH; WATERMAN, 1981), bem como diversas equações e esquemas para exemplificar seu funcionamento. Na Equação 2.4 é apresentada a definição recursiva do algoritmo de Smith-Waterman.

$$\begin{aligned} F(0,0) &= 0 \\ F(i,j) &= max \begin{cases} F(i-1,j-1) + s(s_j^1, s_i^2) \\ F(i-1,j) + d \\ F(i,j-1) + d \\ 0 \end{cases} \end{aligned} \tag{2.4}$$

Onde:

- $\boldsymbol{F(i-1,j-1) + s(s_i^1, s_j^2)}$**:** quando s_i^1 alinha com s_j^2, somando o *score* obtido entre s_i^1 e s_j^2;
- $\boldsymbol{F(i-1,j) + d}$**:** quando s_i^1 alinha com um *gap*;
- $\boldsymbol{F(i,j-1) + d}$**:** quando s_j^2 alinha com um *gap*.

A diferença principal da abordagem adotada pelo algoritmo de Smith-Waterman é a inclusão do escore zero na definição recursiva. Isso significa que a matriz de PD nunca terá

valores negativos e o início e o fim de uma sequência local, serão delimitados pelos valores zeros no início da sequência e o valor zero no fim da sequência. A seguir é apresentada uma simulação utilizando a PD, devendo-se considerar as seguintes sequências e parâmetros para realizar a simulação:

- $s^1 = \text{AGC}$;
- $s^2 = \text{AAG}$;
- $d = -5$ (*gap penalty*);
- $s(s_i^1, s_j^2) = m$ é a função de escore, onde m é um valor obtido cruzando os valores de s_i^1 e s_j^2 em uma matriz de substituição.

A etapa inicial constitui a inicialização da matriz de PD, de acordo com a função recursiva apresentada na Equação 2.4. Diferente do alinhamento global, onde a inicialização da matriz considera o valor de penalidade parametrizado (*gap penalty*), a matriz de alinhamento local, de acordo com a função recursiva, tem as células de *gaps* inicializadas com zero. É importante frisar que no alinhamento local nenhuma célula poderá ter valor menor que zero, pois a função recursiva impede isso. Na Figura 28a é apresentado o estado inicial da matriz de programação dinâmica de alinhamento local.

Figura 28 – (a) Matriz de PD e (b) matriz de substituição para a simulação do algoritmo de Smith-Waterman – Etapa 1

		A	A	G
	0	0	0	0
A	0			
G	0			
C	0			

(a)

Matriz de Substituição				
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

(b)

Fonte: Elaborada pelo autor

Na segunda etapa do algoritmo, os valores de cada célula são processados de acordo com a Equação 2.4. Na Figura 29a é apresentada a simulação do cálculo realizado na célula $(1, 1)$.

Figura 29 – (a) Matriz de PD, (b) matriz de precedência e (c) matriz de substituição para a simulação do algoritmo de Smith-Waterman – Etapa 2

		A	A	G
	0	0	0	0
A	0	2		
G	0			
C	0			

(a)

		A	A	G
A		↘		
G				
C				

(b)

Matriz de Substituição				
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

(c)

Fonte: Elaborada pelo autor

Na Equação 2.5 é apresentado o cálculo considerando a célula $(1, 1)$ da matriz de PD.

$$F(1,1) = max \begin{cases} F(1-1,1-1) + s(s1_1, s2_1) = F(0,0) + 2 = 0 + 2 = 2 \\ F(1-1,1) + d = F(0,1) + (-5) = 0 - 5 = -5 \\ F(1,1-1) + d = F(1,0) + (-5) = 0 - 5 = -5 \\ 0 \end{cases} = 2 \quad (2.5)$$

Pode-se observar na Figura 29b que a seta está direcionada da célula $(0, 0)$ para célula $(1, 1)$. Essa seta indica que o escore máximo foi obtido a partir do cálculo da chamada recursiva $F(i-1, j-1) + s(s_i^1, s_j^2)$.

Na Figura 30 é apresentado o resultado final da simulação. Diferente do alinhamento global, onde o processo de recursividade do algoritmo tem início na última linha e coluna da matriz de programação dinâmica, no alinhamento local, o algoritmo procura pelo maior escore, no caso o valor 4 na célula $(2, 3)$ e retorna até encontrar um zero. Nesse exemplo, é possível observar que existem dois alinhamentos, pois o valor 2 na célula $(1, 1)$ é outro alinhamento possível, visto que ele não foi englobado no alinhamento de maior escore.

Figura 30 – (a) Matriz de PD, (b) matriz de precedência e (c) matriz de substituição para a simulação do algoritmo de Smith-Waterman – Etapa Final

		A	A	G
	0	0	0	0
A	0	2	2	0
G	0	0	0	4
C	0	0	0	0

(a)

		A	A	G
A		↘	↘	
G				↘
C				

(b)

Matriz de Substituição				
	A	C	G	T
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

(c)

Fonte: Elaborada pelo autor

Dessa forma, após a execução final do algoritmo, têm-se os seguintes alinhamentos locais, apresentados na Figura 31.

Figura 31 – Alinhamentos obtidos após a execução do algoritmo de Smith-Waterman

	Alinhamento 1	**Alinhamento 2**
s1 =	**A G**	**A**
s2 =	**A G**	**A**

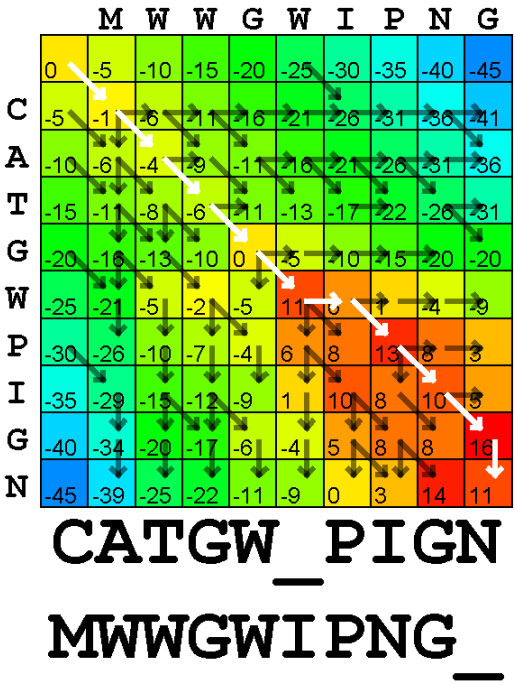
Fonte: Elaborada pelo autor

Analisando as Figuras 30a, 30b e 31 é possível identificar que o alinhamento 1 tem um escore 4 e o alinhamento 2 tem um escore de 2. Nesse cenário, obtiveram-se dois alinhamentos, entretanto, em cenários mais realísticos pode-se obter uma quantidade grande de alinhamentos, cada qual com seu escore. Diante disso, é fundamental que se defina um ponto de corte do algoritmo, ou seja, um valor de escore mínimo. Somente os alinhamentos com escores iguais ou superiores ao mínimo são retornados como resultado do alinhamento executado.

Na Figura 32 e na Figura 33 podem ser vistos, respectivamente, os alinhamentos global e local de duas proteínas hipotéticas de sequências CATGWPIGN e MWWGWIPNG. No alinhamento de proteínas, são usados dois tipos de matrizes de substituição: a matriz *Point Accepted Mutations* (PAM) proposta por Dayhoff, Schwartz e Orcutt (1978) e a matriz *BLOcks SUbstitution Matrix* (BLOSUM) desenvolvida por Henikoff e Henikoff

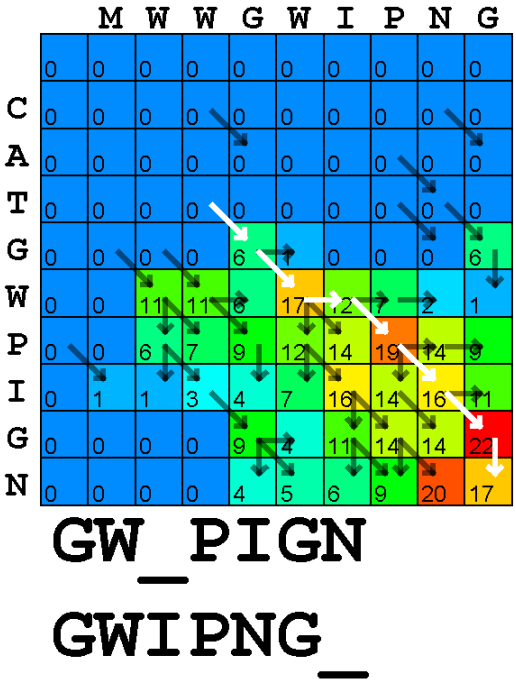
(1992). Na Figura 32 e na Figura 33, as setas pretas indicam os valores da matriz de precedência, enquanto as setas brancas indicam o alinhamento que está sendo exibido abaixo da matriz colorida.

Figura 32 – Exemplo de alinhamento global de duas proteínas hipotéticas



Fonte: Elaborada pelo autor

Figura 33 – Exemplo de alinhamento local de duas proteínas hipotéticas



Fonte: Elaborada pelo autor

2.1.2.3 Implementação Computacional

A seguir são apresentadas duas listagens de código fonte, escritas usando a linguagem de programação Java, que contém o trecho dos algoritmos de alinhamento global e local em que as matrizes de PD e de precedência são calculadas.

- **Alinhamento Global**

```
1  import java.util.ArrayList;
2  import java.util.List;
3
4  /**
5   * Trecho da implementacao do algoritmo de Needleman-Wunsch
6   * @author David Buzatto
7   */
8  public class AlinhamentoGlobal {
9
10     private void calcularMatrizes( String s1, String s2, Function<
           Character, Character, Integer> scoreFunction ) {
11
12         int[][] matrizPD = new int[s1.length()+1][s2.length()+1];
13         List<String>[][] matrizPrecedencia = (List<String>[][]) new
               List[s1.length()+1][s2.length()+1];
14         int d = -5;
15
16         for ( int i = 1; i < s1.length()+1; i++ ) {
17             matrizPD[i][0] = matrizPD[i-1][0] + d;
18         }
19
20         for ( int j = 1; j < s2.length()+1; j++ ) {
21             matrizPD[0][j] = matrizPD[0][j-1] + d;
22         }
23
24         for ( int i = 1; i < matrizPD.length; i++ ) {
25             for ( int j = 1; j < matrizPD[i].length; j++ ) {
26
27                 int r1 = matrizPD[i-1][j-1] + scoreFunction.apply(
                       s1.charAt(i-1), s2.charAt(j-1) );
28                 int r2 = matrizPD[i-1][j] + d;
29                 int r3 = matrizPD[i][j-1] + d;
30                 List<String> caminho = new ArrayList<>();
31
32                 int maior = r1;
33                 if ( maior < r2 ) {
34                     maior = r2;
35                 }
36                 if ( maior < r3 ) {
37                     maior = r3;
```

```
38                  }
39
40                  if ( r1 == maior ) {
41                      caminho.add( "\\" );
42                  }
43                  if ( r2 == maior ) {
44                      caminho.add( "|" );
45                  }
46                  if ( r3 == maior ) {
47                      caminho.add( "-" );
48                  }
49
50                  matrizPD[i][j] = maior;
51                  matrizPrecedencia[i][j] = caminho;
52              }
53
54          }
55      }
56  }
```

- **Alinhamento Local**

```
1  import java.util.ArrayList;
2  import java.util.List;
3
4  /**
5   * Trecho da implementacao do algoritmo de Smith-Waterman.
6   * @author David Buzatto
7   */
8  public class AlinhamentoLocal {
9
10     private void calcularMatrizes( String s1, String s2, Function<
           Character, Character, Integer> scoreFunction ) {
11
12         int[][] matrizPD = new int[s1.length()+1][s2.length()+1];
13         List<String>[][] matrizPrecedencia = (List<String>[][]) new
               List[s1.length()+1][s2.length()+1];
14         int d = -5;
15
16         for ( int i = 1; i < matrizPD.length; i++ ) {
17
18             for ( int j = 1; j < matrizPD[i].length; j++ ) {
19
20                 int r1 = matrizPD[i-1][j-1] + scoreFunction.apply(
                       s1.charAt(i-1), s2.charAt(j-1) );
21                 int r2 = matrizPD[i-1][j] + d;
22                 int r3 = matrizPD[i][j-1] + d;
23                 List<String> caminho = new ArrayList<>();
```

```
24
25                  int maior = 0;
26                  if ( maior < r1 ) {
27                      maior = r1;
28                  }
29                  if ( maior < r2 ) {
30                      maior = r2;
31                  }
32                  if ( maior < r3 ) {
33                      maior = r3;
34                  }
35
36                  if ( r1 == maior ) {
37                      caminho.add( "\\" );
38                  }
39                  if ( r2 == maior ) {
40                      caminho.add( "|" );
41                  }
42                  if ( r3 == maior ) {
43                      caminho.add( "-" );
44                  }
45
46                  matrizPD[i][j] = maior;
47                  matrizPrecedencia[i][j] = caminho;
48              }
49
50          }
51      }
52  }
```

A partir da ideia da comparação/alinhamento das sequências das estruturas primárias das proteínas, um problema unidimensional, visto a característica das sequências, pode-se então extrapolar essa ideia, mudando do domínio estrutural unidimensional para o domínio estrutural tridimensional ao se comparar a conformação de duas proteínas distintas. Na próxima seção esse tema será explorado.

2.1.3 Comparação ou Alinhamento Estrutural

A comparação ou alinhamento da conformação das proteínas é importante, pois pode ser usada como ferramenta para verificar as similaridades e/ou diferenças entre estruturas de duas proteínas, permitindo assim inferir funções biológicas análogas, visto que, como já apresentado, a similaridade estrutural implica, na maioria das vezes, em similaridade funcional e relacionamento evolutivo entre as proteínas comparadas (LESK, 2016), algo que pode não ser detectado utilizando apenas as informações das estruturas primárias das

proteínas (KOEHL, 2001). Além disso, a tarefa de comparar a estrutura de proteínas é importante, pois há interesse, por parte dos biólogos evolucionários, em estabelecer relações evolutivas entre proteínas, utilizando para isso suas estruturas (MURZIN, 1996; MURZIN, 1998); por parte dos físicos, ao estudar padrões estruturais comuns com o objetivo de obter regras que ajudem em um melhor entendimento da arquitetura das proteínas (CHOTHIA; FINKELSTEIN, 1990); por parte dos pesquisadores da genômica estrutural, em inferir a função de uma proteína que teve apenas a sua estrutura hipotética calculada (HWANG et al., 1999); e, contribuir com o cálculo de acuracidade nos preditores estruturais de proteínas (MOULT et al., 1999).

Para realizar esse tipo de comparação, existe a necessidade de se utilizar modelos matemáticos e computacionais capazes de processar toda a estrutura das proteínas que serão comparadas, de modo a verificar se existem similaridades entre as duas proteínas e, caso existam, quais são essas similaridades e em qual ou quais localizações das estruturas elas se encontram. Sendo assim, nesta seção serão apresentadas as principais técnicas capazes de realizar esta tarefa. Na Tabela 3 estão listadas as principais ferramentas computacionais desenvolvidas com o objetivo de comparar estruturas de proteínas. Em seguida, cada uma dessas ferramentas e/ou algoritmos serão detalhadas, bem como toda a teoria que embasa seus respectivos funcionamentos.

Tabela 3 – Principais Ferramentas Computacionais para Comparação Estrutural de Proteínas

Nome	**Endereço na Web**
STAMP (RUSSELL; BARTON, 1992)	<http://www.compbio.dundee.ac.uk/manuals/stamp.4.2/>
Método Dali - DaliLite e DaliServer (HOLM; SANDER, 1993) (HOLM; SANDER, 1995) (HOLM; PARK, 2000) (HASEGAWA; HOLM, 2009) (HOLM; ROSENSTROM, 2010) (HOLM; LAAKSO, 2016)	<http://ekhidna2.biocenter.helsinki.fi/dali/>
CE (SHINDYALOV; BOURNE, 1998) (SHINDYALOV; BOURNE, 2001) (GUDA et al., 2004) (JIA et al., 2004)	<http://cl.sdsc.edu/>
MATRAS (KAWABATA; NISHIKAWA, 2000) (KAWABATA, 2003)	<http://strcomp.protein.osaka-u.ac.jp/matras/>
FATCAT (YE; GODZIK, 2003) (YE; GODZIK, 2004)	<http://fatcat.burnham.org/>

Fonte: Elaborada pelo autor

Apesar do funcionamento dos algoritmos de alinhamento estrutural ser diferente, a maioria deles compartilha o uso de alguns cálculos para se obter valores de distância

relativa entre as proteínas alinhadas, significância estatística dos alinhamentos calculados, entre outros. Na lista a seguir são apresentadas as medidas utilizadas pela maioria dos algoritmos:

- **RMSD:** A medida de *Root-Mean-Square Distance* ou *Root-Mean-Square Deviation* (RMSD), de acordo com Carugo e Pongor (2001), é usada para calcular a distância entre os átomos equivalentes em duas estruturas tridimensionais distintas. Essa medida é dada pela Equação 2.6.

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{n} d_i^2}{n}} \tag{2.6}$$

  Onde:

  - i é o i-ésimo átomo das estruturas;
  - n é a quantidade de átomos das estruturas;
  - d é a distância entre os átomos i equivalentes das duas estruturas em questão.

  Segundo Maiorov e Crippen (1994), quanto maior o valor do RMSD, mais dissimilares são as estruturas e, quanto menor, mais similares são, sendo que um RMSD de valor mínimo, ou seja, igual a 0, significa que as duas estruturas comparadas são iguais, pois têm distância igual a 0.

- ***z-score*:** O *z-score*, escore z, valor padronizado, ou escore padronizado, é o valor associado com a i-ésima observação de uma variável x. Em outras palavras, o *z-score* representa a quantidade de desvios padrão que um dado de uma amostra se encontra acima ou abaixo da média (TRIOLA, 2014). O *z-score* é dado pela Equação 2.7.

$$z_i = \frac{x_i - \bar{x}}{\sigma} \tag{2.7}$$

  Onde:

  - z_i é *z-score* do valor de um dado da amostra;
  - x_i é o valor de um dado da amostra;
  - $\bar{x}$ é a média da amostra;
  - σ é o desvio padrão da amostra.

  Por exemplo, para uma distribuição normal de $\bar{x} = 7$ e $\sigma = 1,6$, um $x_i = 8,6$ teria $z_i = 1$. Para $x_i = 7$, $z_i = 0$, enquanto para $x_i = 3,8$, $z_i = -2$.

- ***p-value*:** O *p-value*, ou valor p, ou valor de probabilidade, representa a probabilidade de que, tomando a hipótese nula como verdadeira, um resultado de um teste seja verdadeiro (TRIOLA, 2014), sendo que, de acordo com Wrabl e Grishin (2008), a quantificação desse valor de significância estatística é essencial para viabilizar a interpretação do resultado de uma comparação de similaridade entre estruturas de duas proteínas. Por exemplo, assumindo que duas proteínas distintas são similares (hipótese nula), o *p-value* para essa comparação será um valor alto, maior que 0,05 (5%), caso realmente as proteínas sejam similares, enquanto um valor menor ou igual a 0,05 indica que provavelmente as duas proteínas não são similares. De modo geral, os critérios de decisão ao se usar o *p-value* são dados, segundo Triola (2014), pelas condições abaixo:

$$\begin{array}{ll} \text{Se } p \leq \alpha, & \text{rejeite } H_0 \\ \text{Se } p > \alpha, & \text{deixe de rejeitar } H_0 \end{array} \tag{2.8}$$

  Onde:

  - p é o *p-value*;
  - α é o nível de significância, usualmente 0,05;
  - H_0 é a hipótese nula.

Sendo assim, nas próximas cinco seções serão apresentados os algoritmos e/ou modelos matemáticos apresentados na Tabela 3.

2.1.3.1 STAMP

O algoritmo de alinhamento estrutural de proteínas *STructual Alignment of Multiple Proteins* (STAMP) foi desenvolvido por Russell e Barton (1992) e será detalhado nesta seção. Para executar os três passos que o compõe, detalhados a seguir, faz uso de três técnicas:

1. ***Least-Squares Fitting:*** A ideia desta técnica é embasada em encontrar a melhor curva que superpõe/casa (*fit*) com um conjunto de pontos (LEAST..., 2016) sendo que, no algoritmo STAMP, para duas proteínas A e B, define-se dois conjuntos de n pontos (x, y, z) no espaço cartesiano que representam os átomos da proteína A e da proteína B e calculam-se transformações sucessivas (translação e rotação) nesses conjuntos de modo a diminuir o RMSD, expresso na forma:

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{n} \left((x_{A_i} - x_{B_i})^2 + (y_{A_i} - y_{B_i})^2 + (z_{A_i} - z_{B_i})^2 \right)}{n}} \tag{2.9}$$

2. **Análise de Agrupamentos Hierárquicos (*Hierarchical Cluster Analysis*):** Para N objetos com um escore gerado a partir da comparação com os $N(N-1)/2$ possíveis pares de objetos, este método gera uma árvore (dendograma) que organiza esses objetos de forma hierárquica, agrupando objetos com maior similaridade nos ramos mais altos, permitindo que estruturas com maior similaridade sejam processadas primeiro, visto que o nível da árvore que as mesmas se encontram é mais raso;

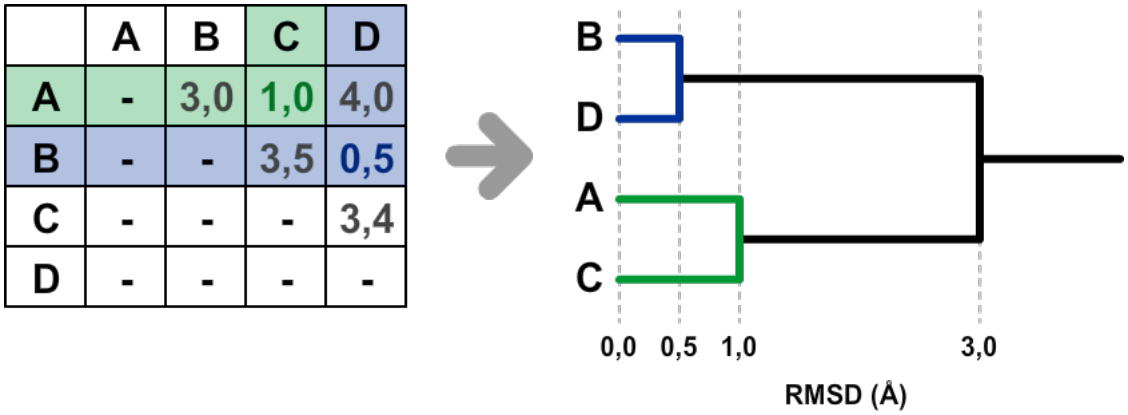
3. **Programação Dinâmica:** No método STAMP, a técnica de PD, já apresentada na seção "Comparação ou Alinhamento de Sequências", foi adaptada para processar dados tridimensionais no alinhamento das estruturas, ao passo que para os alinhamentos de sequências ela é usada para processar as sequências das estruturas primárias das proteínas, que por sua vez têm natureza unidimensional.

O funcionamento do algoritmo STAMP, utilizando as técnicas apresentadas anteriormente, como já dito, é composto de três passos:

1. **Superposição Inicial:** Para as estruturas tridimensionais de N proteínas, é feito, primeiramente, o alinhamento múltiplo de sequências utilizando o algoritmo de Barton e Sternberg (1987). Após o alinhamento, as regiões sem *gaps* dos M alinhamentos obtidos são estruturalmente superpostas, utilizando para isso os C^{α} dos resíduos alinhados. Para cada um dos $N(N-1)/2$ pares de proteínas, os M átomos equivalentes são comparados usando a técnica de *Least-Squares Fitting*, gerando assim valores de RMSD para cada comparação. A árvore de similaridade entre as regiões similares encontradas é então gerada, com base nos valores de RMSD. Utilizando essa árvore, é então obtida a superposição estrutural múltipla das N proteínas que estão sendo comparadas;

   Na Figura 34 pode ser visto um esquema ilustrativo da derivação da árvore de similaridade (direita), que é construída a partir da matriz de similaridade (esquerda), obtida após o alinhamento múltiplo das sequências de quatro proteínas hipotéticas A, B, C e D. A superposição das proteínas do exemplo da Figura 34 é obtida ao se emparelhar a proteína B com a D (similaridade de 0,5Å), em seguida emparelhar a proteína A com a C (similaridade de 1,0Å) e, por fim, emparelhando os dois pares gerados, BD e AC, obtendo a superposição inicial;

Figura 34 – Esquema ilustrativo da superposição inicial do algoritmo STAMP



Fonte: Adaptado de Russell e Barton (1992) pelo autor

2. **Alinhamento Múltiplo de Estruturas:** Apesar da superposição gerada no primeiro passo ser a melhor superposição possível ao se utilizar a árvore de similaridade obtida, quaisquer erros que foram gerados durante o alinhamento múltiplo das sequências das proteínas são propagados para a árvore. Sendo assim, existe a necessidade de melhorar a superposição inicial, utilizando para isso um critério estrutural que dita qual a probabilidade de dois resíduos, tomados de duas estruturas distintas, sejam equivalentes. Para isso, utiliza-se a Equação 2.10:

$$P_{ij} = \left\{ \exp -\frac{d_{ij}^2}{2E_1^2} \right\} \left\{ \exp -\frac{s_{ij}^2}{2E_2^2} \right\} \tag{2.10}$$

Onde:

- P_{ij} é a probabilidade que os resíduos i e j, de duas proteínas distintas, têm de serem estruturalmente equivalentes;
- E_1 e E_2 são constantes;
- d_{ij} é a distância entre os C^{α};
- s_{ij} é a medida de similaridade e é calculada usando a Equação 2.11

$$\begin{aligned} s_{ij}^2 = \{ &(\Delta x_{ij} - \Delta x_{i-1,j-1})^2 + (\Delta y_{ij} - \Delta y_{i-1,j-1})^2 + (\Delta z_{ij} - \Delta z_{i-1,j-1})^2 + \\ &(\Delta x_{ij} - \Delta x_{i+1,j+1})^2 + (\Delta y_{ij} - \Delta y_{i+1,j+1})^2 + (\Delta z_{ij} - \Delta z_{i+1,j+1})^2 \} \end{aligned} \tag{2.11}$$

Na Equação 2.10, o primeiro termo, relacionado à distância, contribui para a medida de proximidade espacial dos dois resíduos e o segundo termo, relacionado à similaridade, afeta a definição da similaridade conformacional entre os mesmos dois resíduos. As contantes E_1 e E_2 afetam o quanto cada termo contribuirá para a obtenção da probabilidade, sendo que quando $E_1 > E_2$ a probabilidade será mais

relacionada à similaridade conformacional do que à distância espacial e quando $E_1 < E_2$ a probabilidade será mais relacionada à distância espacial do que à similaridade conformacional. Para duas estruturas distintas de, respectivamente, m e n resíduos, será gerada uma matriz m por n com as probabilidades de cada par de resíduos i e j. Essa matriz de probabilidade é então usada para obter o caminho do alinhamento ótimo, usando a mesma estratégia do algoritmo de alinhamento local de Smith e Waterman (1981). Este processo de alinhamento/refinamento é repetido iterativamente até que a diferença entre o escore do alinhamento anterior e o escore do alinhamento atual seja menor ou igual à 0,1% do escore do alinhamento anterior.

3. **Normalização:** Dependendo da família das proteínas que são alinhadas, os valores de P_{ij} (Equação 2.10) têm uma variação que pode ser observada. Em alinhamentos par-a-par entre proteínas de famílias diferentes, os valores de P_{ij} são equivalentes em regiões com estruturas secundárias, entretanto a média dos valores de P_{ij} diminui ao passo que aumenta a quantidade de estruturas alinhadas, assim como o desvio padrão. No experimento realizado por Russell e Barton (1992), em que se pode verificar essa dependência entre média e desvio padrão em relação ao comprimento do alinhamento (quantidade de estruturas alinhadas), tanto a média quanto o desvio padrão foram ajustados para, respectivamente, 0,020 e 0,10, ou seja, $\bar{x}_t = 0,020$ e $\sigma_t = 0,10$, de modo a padronizar as medidas para quaisquer comparações. A correção dessa dependência, para alinhamentos múltiplos, é feita usando as relações exponenciais apresentadas nas Equações 2.12 e 2.13.

$$\bar{x}_{\text{par-a-par}} = \exp(-0,950 \log(L) + 0,686) \tag{2.12}$$

$$\sigma_{\text{par-a-par}} = \exp(-0,474 \log(L) + 0,0152) \tag{2.13}$$

Onde:

- L é o tamanho médio do alinhamento das sequências que estão sendo processadas.

As relações apresentadas nas Equações 2.14 e 2.15 são usadas, respectivamente, para corrigir os valores para os alinhamentos múltiplos e par-a-par.

$$\bar{x}_c = \bar{x}_t \left(\frac{\bar{x}_{\text{múltiplo}}}{\bar{x}_{\text{par-a-par}}} \right) \tag{2.14}$$

$$\sigma_c = \sigma_t \left(\frac{\sigma_{\text{múltiplo}}}{\sigma_{\text{par-a-par}}} \right) \tag{2.15}$$

Os valores para P'_{ij} são obtidos a partir da Equação 2.14.

$$P'_{ij} = \left(\frac{P_{ij} - \bar{x}_c}{\sigma_c} \right) \tag{2.16}$$

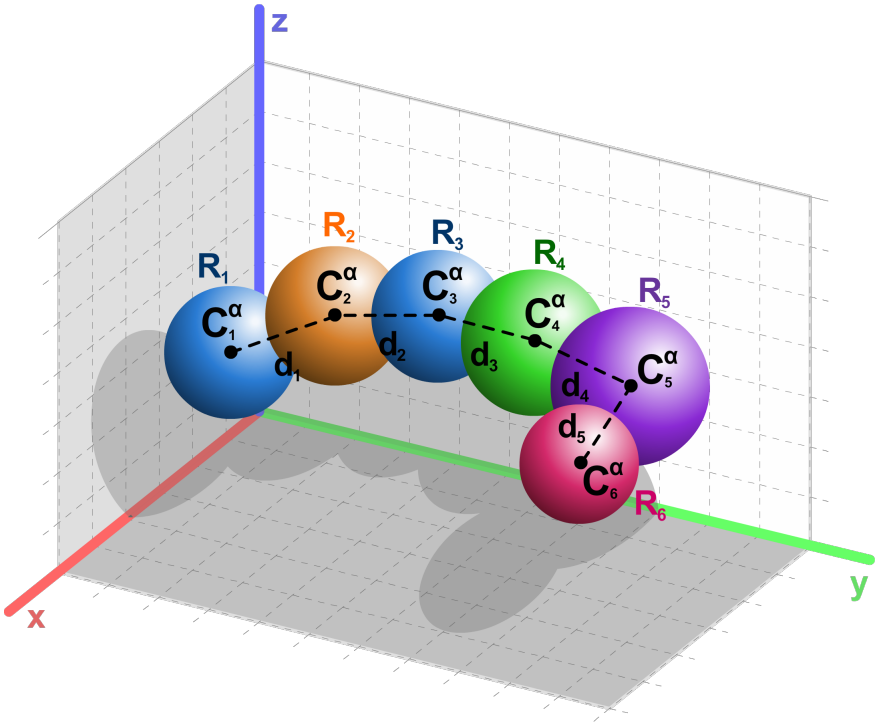
Pelo exposto, o algoritmo STAMP, desenvolvido por Russell e Barton (1992), utiliza dados das sequências (estruturas primárias) e dos modelos tridimensionais das proteínas para obter o alinhamento final de duas ou mais proteínas. Existem também algoritmos que focam em alinhamentos de somente duas proteínas, executando para isso outras técnicas. O STAMP é utilizado no *plug-in* MultiSeq (EARGLE; WRIGHT; LUTHEY-SCHULTEN, 2006; ROBERTS et al., 2006) do *software* de visualização de estruturas de proteínas VMD (HUMPHREY; DALKE; SCHULTEN, 1996). A implementação original do STAMP, bem como sua documentação, pode ser encontrada na Web a partir do endereço <http://www.compbio.dundee.ac.uk/manuals/stamp.4.2/>.

2.1.3.2 Método Dali: Comparação por Alinhamento de Matrizes de Distância

Uma matriz de distância (*distance matrix*), também denominada mapa de distância (*distance map*) ou gráfico de distância (*distance plot*) é uma representação em Duas Dimensões (2D) de uma estrutura em Três Dimensões (3D) que pode ser utilizada para descrever e comparar conformações de proteínas (HOLM; SANDER, 1993). De acordo com Holm e Sander (1993), o tipo de matriz de distância utilizada para a comparação estrutural de proteínas é normalmente a que codifica as distâncias, par-a-par, entre os centros dos resíduos, isto é, a distância entre os C^{α} de cada resíduo.

Na Figura 35 é apresentada uma representação de uma proteína hipotética que possui seis resíduos, ilustrados como esferas etiquetadas com a letra “R”, com seus centros, representados pelos C^{α}, ligados por segmentos de reta tracejados que, por sua vez, representam as distâncias entre os centros de cada resíduo e são denotadas pela letra “d”.

Figura 35 – Exemplo de estrutura 3D de uma proteína hipotética de seis resíduos para obtenção da matriz de distância



Fonte: Elaborada pelo autor

A matriz de distância do esquema apresentado na Figura 35 é mostrada na Figura 36, sendo que para os cálculos das distâncias foram considerados os seguintes valores:

- $d_1 = 1$Å
- $d_2 = 1,2$Å
- $d_3 = 1,1$Å
- $d_4 = 2$Å
- $d_5 = 1$Å

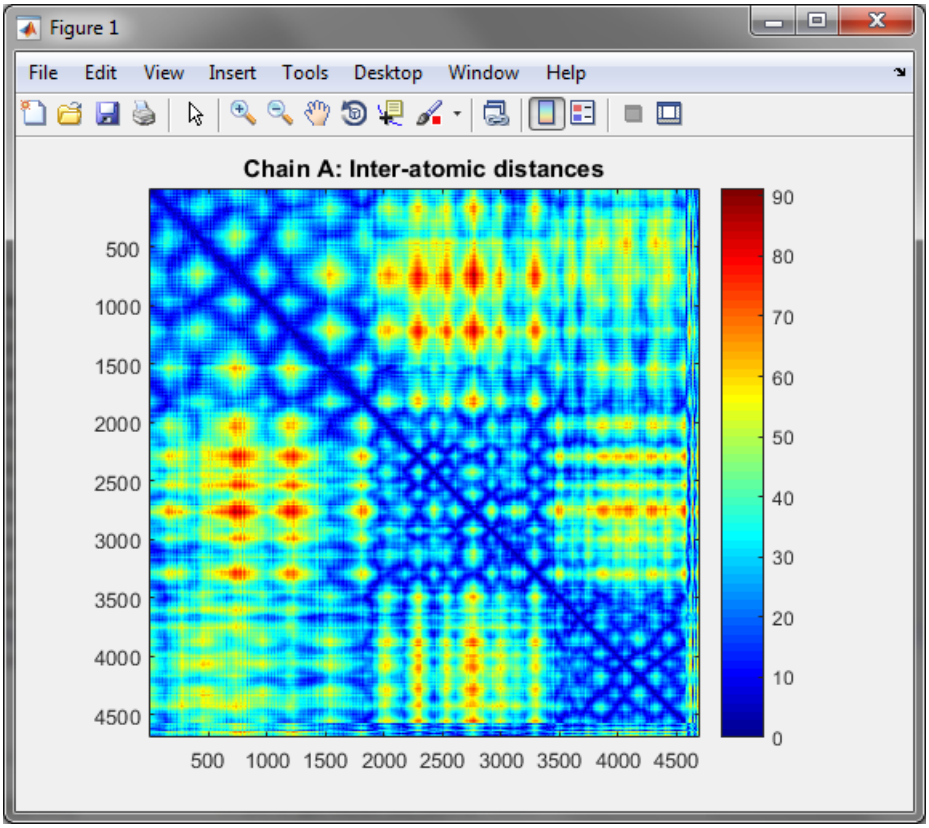
Figura 36 – Matriz de distância, com valores em Ångström, obtida a partir do modelo 3D de uma proteína hipotética de seis resíduos

	C_1^α	C_2^α	C_3^α	C_4^α	C_5^α	C_6^α
C_1^α	0	1	2,2	3,3	5,3	6,3
C_2^α	1	0	1,2	2,3	4,3	5,3
C_3^α	2,2	1,2	0	1,1	3,1	4,1
C_4^α	3,3	2,3	1,1	0	2	3
C_5^α	5,3	4,3	3,1	2	0	1
C_6^α	6,3	5,3	4,1	3	1	0

Fonte: Elaborada pelo autor

Na Figura 37 pode ser vista a matriz de distância de todos os átomos da proteína Cry1Aa1 (PDB: 1CIY). Neste gráfico representativo da matriz, pode-se ver que as distâncias entre os átomos está representada utilizando uma escala de cores, variando de 0 (azul) a 90 (vermelho), sendo que a unidade de medida utilizada é o Ångström.

Figura 37 – Matriz de distância de todos os átomos da proteína Cry1Aa1 (PDB: 1CIY)



Fonte: Elaborada pelo autor utilizando o *software* MATLAB versão R2015b

O método de comparação de estruturas de proteínas utilizando o alinhamento de matrizes de distância, denominado Método Dali, foi desenvolvido por Holm e Sander (1993), sendo que este método utiliza a ideia de calcular as matrizes de distância de duas proteínas que serão comparadas, alinhar tais matrizes e então maximizar a sobreposição

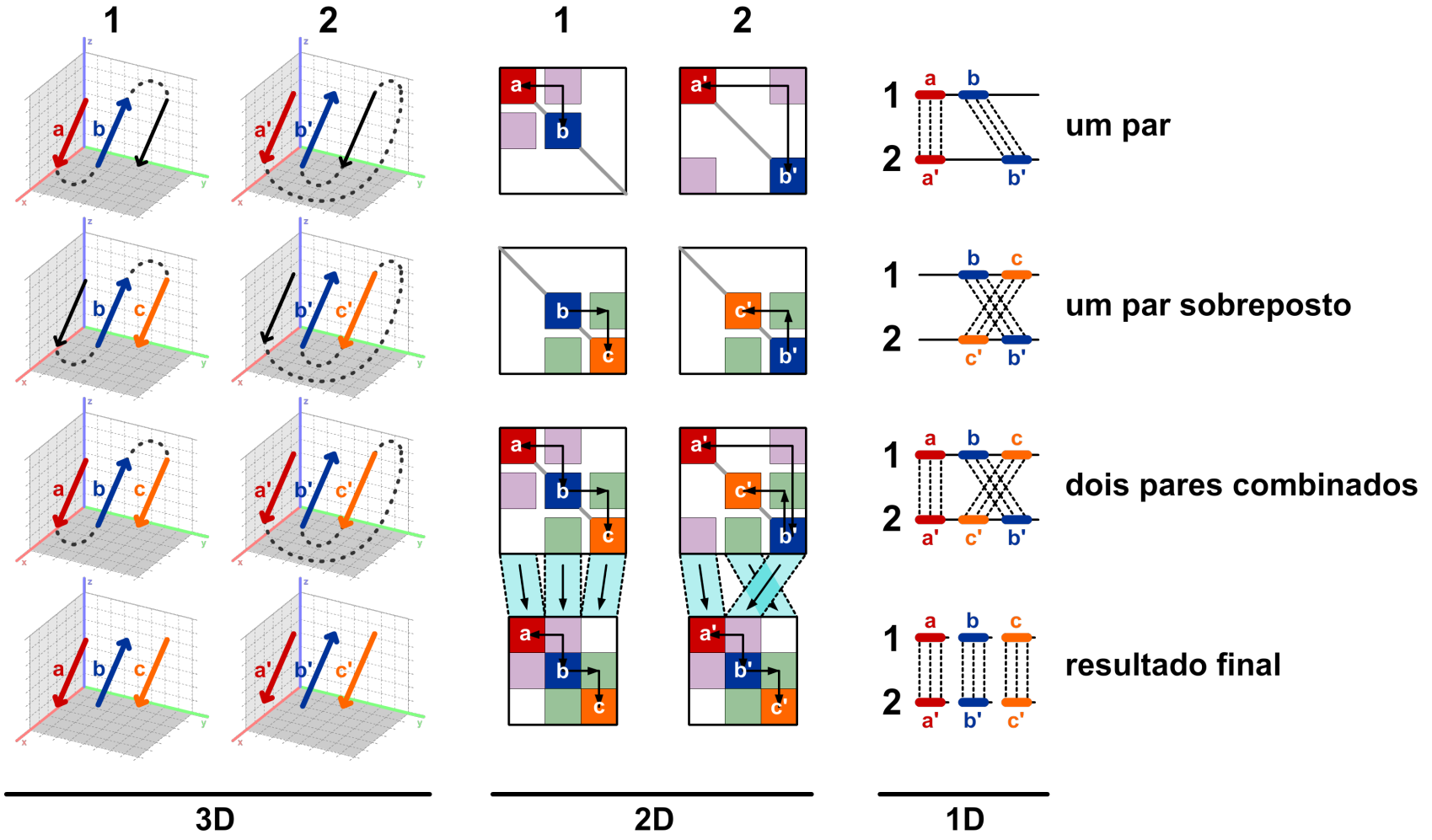
das regiões similares entre as duas proteínas que estão sendo comparadas. Nessa seção este método será descrito com base no trabalho de Holm e Sander (1993).

Na Figura 38 é apresentado um diagrama ilustrativo do funcionamento do algoritmo. O diagrama é dividido em três seções:

- **3D:** estruturas tridimensionais das proteínas;
- **2D:** matrizes de distância das proteínas;
- **1D:** estruturas primárias das proteínas.

Neste diagrama é apresentada a comparação de duas proteínas hipotéticas que possuem organização topológica distinta e que são formadas por uma folha β composta de três conformações β. As conformações β equivalentes são representadas pelas letras a, b e c na proteína 1 e a', b' e c' na proteína 2, utilizando as cores vermelho (a, a'), azul (b, b') e laranja (c, c'). Nas matrizes de distância, os padrões de contato similares são representados por quadrados de cores roxo e verde, enquanto os quadrados que estão na diagonal da matriz representam as respectivas conformações β de cada proteína, coloridos de acordo com as cores atribuídas a cada conformação. O cálculo do escore de similaridade é feito utilizando todas as diferenças par-a-par das duas matrizes de distância. Na primeira linha do diagrama, o alinhamento é iniciado, casando os padrões de contato (a,b) e (a',b'), representados em roxo nas matrizes. Na segunda linha, os fragmentos b e b' são utilizados para se buscar novos fragmentos, estendendo assim o alinhamento. Os fragmentos c e c' são identificados, visto que os padrões de contato (b,c) e (b',c'), em verde nas matrizes, são similares. Na terceira linha, os padrões (a,b)-(a',b') e (b,c)-(b',c') são fundidos gerando o alinhamento (a,b,c)-(a',b',c'). Na última linha é apresentado o alinhamento final, após a remoção de inserções/remoções e da reorganização dos segmentos b' e c' na proteína 2. O alinhamento final em Uma Dimensão (1D) é apresentado no canto inferior direito do diagrama. Essa comparação de padrões de contato que é utilizada é independente das lacunas dos alinhamentos, as inserções/remoções mencionadas anteriormente, e também é capaz de identificar casamentos em sequências de direção invertida (HOLM; SANDER, 1993).

Figura 38 – Esquema ilustrativo do funcionamento do algoritmo de comparação de estruturas de proteínas usando matrizes de distância



Fonte: Adaptado de Holm e Sander (1993) pelo autor

Neste método de comparação, os escores de similaridade são calculados com base em três definições:

1. **Escore de Similaridade Aditiva:** Considerando duas proteínas A e B, o escore do alinhamento de duas subestruturas é calculado utilizando a Equação 2.17.

$$S = \sum_{i=1}^{L} \sum_{j=1}^{L} \phi\left(i, j\right) \tag{2.17}$$

   Onde:

   - i e j são os índices dos intervalos dos pares equivalentes, isto é, $i = (i_A, i_B)$ e $j = (j_A, j_B)$;
   - L é a quantidade de pares combinados, ou seja, o tamanho de cada subestrutura;
   - ϕ é a medida de similaridade baseada em alguma relação de emparelhamento, neste caso, as distâncias $\mathrm{C}^{\alpha}-\mathrm{C}^{\alpha}$ (d_{ij}^{A} e d_{ij}^{B}). Nas Equações 2.18 e 2.19 são apresentadas outras duas medidas de similaridade;
   - Para uma dada forma funcional $\phi\left(i, j\right)$, o maior valor de S corresponde à equivalência ótima em um conjunto de resíduos.

2. **Escore de Similaridade Rígida:** Define a similaridade entre dois requisitos contraditórios que são: a maximização de resíduos equivalentes e a minimação de desvios estruturais. Este escore é obtido utilizando a Equação 2.18.

$$\phi^{R}\left(i, j\right) = \theta^{R} - \left|d_{ij}^{A} - d_{ij}^{B}\right| \tag{2.18}$$

   Onde:

   - O sobrescrito R denota a rigidez;
   - d_{ij}^{A} e d_{ij}^{B} são os elementos equivalentes das matrizes de distância das proteínas A e B;
   - $\theta^{R} = 1,5$Å representa o menor nível de similaridade.

3. **Escore de Similaridade Elástica:** Para se obter as variações relativas entre distâncias, o escore de similaridade elástica é utilizado, tornando-as mais tolerante às distorções geométricas que aparecem gradualmente durante o processo de alinhamento. Este escore é obtido utilizando a Equação 2.19.

$$\phi^{E}\left(i, j\right) = \left\{ \begin{array}{cc} \left(\theta^{E} - \frac{\left|d_{ij}^{A} - d_{ij}^{B}\right|}{d_{ij}^{*}}\right) w\left(d_{ij}^{*}\right), & i \neq j \\ \theta^{E}, & i = j \end{array} \right\} \tag{2.19}$$

   Onde:

- O sobrescrito E denota a elasticidade;
- d^*_{ij} é a média de distância entre d^A_{ij} e d^B_{ij};
- θ^E é o limiar de similaridade, neste caso $\theta^E = 0,20$Å, isto é, 20% de variação. Como exemplo, duas conformações β adjacentes em uma folha β, que têm distância típica entre 4 e 5Å, devem se corresponder em torno de 1Å, enquanto de 2 a 3Å de deslocamento são tolerados para contatos do tipo conformação β-hélice e hélice-hélice, pois possuem distância típica entre 8 a 15Å;
- Pares em faixas de distância grandes são abundantes e menos discriminativos, sendo assim suas contribuições são ponderadas para baixo utilizando a função envelope $w(r) = \exp(-r^2/\alpha^2)$, onde $\alpha = 20$Å, calibrado utilizando o tamanho de um domínio típico.

O algoritmo guloso[14], desenvolvido por Holm e Sander (1993), e que executa o alinhamento das estruturas de duas proteínas, é composto de duas etapas:

1. **Decomposição das Matrizes de Distância:** Nesta etapa todos os padrões de contato das duas matrizes são comparados par-a-par de forma sistemática, sendo que esses padrões de contato são na forma hexapeptídeo-hexapeptídeo, isto é, $(i_A, \ldots, i_{A+5}, j_A, \ldots, j_{A+5})$ na proteína A pareado com $(i_B, \ldots, i_{B+5}, j_B, \ldots, j_{B+5})$ na proteína B, onde o hexapeptídeo $i_A, \ldots, i_{A+5}$ é equivalente a $j_A, \ldots, j_{A+5}$ e o hexapeptídeo $i_B, \ldots, i_{b+5}$ é equivalente a $j_B, \ldots, j_{B+5}$. Padrões similares são armazenados em uma lista chamada de "lista de pares";
2. **Incremento dos Alinhamentos**: Nesta etapa, a lista de pares gerada da etapa anterior é usada de modo a encontrar novos conjuntos de alinhamentos que sejam mais consistentes, sendo que para isso as pontuações dos alinhamentos precisam ser maximizadas utilizando a Equação 2.17.

O Método Dali de Holm e Sander (1993) foi originalmente implementado computacionalmente utilizando a linguagem de programação FORTRAN (IBM *Mathematical FORmula TRANslation System*), sendo que a implementação atual pode ser encontrada na Web a partir do endereço <http://ekhidna2.biocenter.helsinki.fi/dali/>. As ferramentas que possibilitam a utilização do método Dali têm sido atualizadas desde sua concepção, sendo reescritas e compiladas em diversos sistemas operacionais como Linux e Irix, sendo que essas atualizações e melhorias são apresentadas em Holm e Sander (1995), Holm e Park (2000), Hasegawa e Holm (2009), Holm e Rosenstrom (2010) e Holm e Laakso (2016).

[14] Algoritmos gulosos (*greedy algorithms*) são empregados em problemas de otimização. Estes algoritmos sempre escolhem a opção que parece melhor para a solução de um subproblema, ou seja, a solução ótima local, não se preocupando se esta opção levará à solução ótima do problema geral, ou seja, a solução ótima global.

2.1.3.3 CE: Alinhamento por Extensão Combinatória

O algoritmo de *Combinatorial Extension*[15] (CE) para o alinhamento estrutural de duas proteínas, desenvolvido por Shindyalov e Bourne (1998), tem como objetivo diminuir a quantidade de processamento envolvido durante o alinhamento, minimizando, para isso, a quantidade de regiões que serão comparadas entre as duas estruturas. Nesse algoritmo, detalhado nessa seção com base no trabalho de Shindyalov e Bourne (1998), assume-se que o caminho percorrido para se alinhar cada estrutura, incluindo *gaps* que podem ter sido inseridos, é contínuo e que não existe uma correspondência estrutural ótima. Ainda, esse algoritmo tem como característica não alinhar similaridades não topológicas, ou seja, similaridades em que a ordem dos fragmentos de polipeptídeos não seguem a mesma ordem nas sequências. A seguir são apresentados os passos do algoritmo CE segundo Shindyalov e Bourne (1998).

1. **Definição do Caminho do Alinhamento:** O alinhamento das estruturas de duas proteínas A e B, de tamanho n^A e n^B, respectivamente, é dado pelo maior caminho contínuo P formado por *Aligned Fragment Pairs* (AFPs)[16] de tamanho m, em uma matriz de similaridade S, de dimensões $(n^A - m) \times (n^B - m)$, que representa todos os AFPs que estão em conformidade com o critério de similaridade estrutural, neste caso, superposição estrutural de corpos rígidos e distâncias inter-resíduos. Para cada dois AFPs consecutivos i e $i+1$ no caminho do alinhamento p, uma das três condições a seguir deve ser satisfeita:

$$p_{i+1}^A = p_i^A + m \wedge p_{i+1}^B = p_i^B + m \tag{2.20}$$

ou

$$p_{i+1}^A > p_i^A + m \wedge p_{i+1}^B = p_i^B + m \tag{2.21}$$

ou

$$p_{i+1}^A = p_i^A + m \wedge p_{i+1}^B > p_i^B + m \tag{2.22}$$

Onde:

- p_i^A é a posição do resíduo inicial do AFP na proteína A que se encontra na i-ésima posição do caminho do alinhamento;
- p_i^B é a posição do resíduo inicial do AFP na proteína B que se encontra na i-ésima posição do caminho do alinhamento;
- Na Equação 2.20 são descritos dois AFPs consecutivos alinhados sem *gaps*;

[15] Extensão Combinatória.
[16] Pares de Fragmentos Alinhados.

- Na Equação 2.21 são descritos dois AFPs consecutivos alinhados com inserção de *gaps* na proteína A;
- Na Equação 2.22 são descritos dois AFPs consecutivos alinhados com inserção de *gaps* na proteína B.

2. **Extensão Combinatória do Caminho do Alinhamento:** O caminho do alinhamento é construído utilizando AFPs de tamanho fixo m, sendo que o valor 8, segundo Shindyalov e Bourne (1998), é adotado por ser o que apresenta melhores resultados empiricamente. Sendo assim, um fragmento de comprimento m da proteína A forma um par com um fragmento de comprimento m da proteína B, caso eles satisfaçam o critério de similaridade apresentado no item "Heurísticas para Avaliação de Similaridade e Extensão de Caminho" a seguir. O primeiro AFP que inicia o caminho pode ser selecionado em qualquer posição da matriz de similaridade S. AFPs consecutivos vão sendo adicionados de modo a satisfazer as Equações 2.20, 2.21 e 2.22. Para limitar o tamanho dos *gaps*, as Equações 2.21 e 2.22 são melhoradas com a adição de mais duas condições, respectivamente:

$$p_{i+1}^{A} \leq p_{i}^{A} + m + G \tag{2.23}$$

e

$$p_{i+1}^{B} \leq p_{i}^{B} + m + G \tag{2.24}$$

Onde G é o tamanho máximo permitido para um *gap*, determinado como 30, empiricamente, segundo Shindyalov e Bourne (1998).

3. **Heurísticas para Avaliação de Similaridade e Extensão de Caminho:** Shindyalov e Bourne (1998) definem três medidas para a avaliação de similaridade:

   (a) Distância D_{ij} calculada usando um conjunto independente de distâncias inter-resíduos, onde cada resíduo participa uma e somente uma vez no conjunto de distância selecionado:

$$\begin{aligned} D_{ij} = & \frac{1}{m}\left(\left|d^{A}_{p_i^A p_i^A} - d^{B}_{p_i^B p_i^B}\right| + \left|d^{A}_{p_i^A+m-1,p_j^A+m-1} - d^{B}_{p_i^B+m-1,p_j^B+m-1}\right| + \right. \\ & \left. \sum_{k=1}^{m-2}\left|d^{A}_{p_i^A+k,p_j^A+m-l-k} - d^{B}_{p_i^B+k,p_j^B+m-l-k}\right|\right) \end{aligned} \tag{2.25}$$

   (b) Distância D_{ij} calculada usando o conjunto completo de distâncias inter-resíduos, onde todas as distâncias possíveis são avaliadas, exceto de resíduos vizinhos:

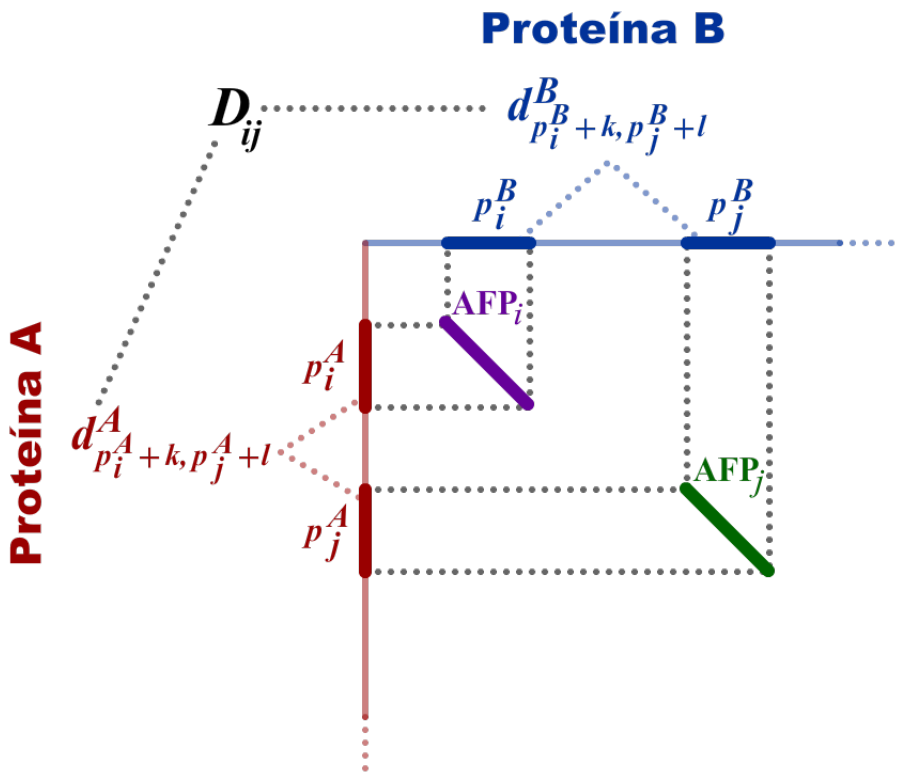
$$D_{ij} = \frac{1}{m^2}\left(\sum_{k=0}^{m-1}\sum_{l=0}^{m-1}\left|d^{A}_{p_i^A+k,p_j^A+l} - d^{B}_{p_i^B+k,p_j^B+l}\right|\right) \tag{2.26}$$

(c) RMSD obtido de estruturas otimamente superpostas como corpos rígidos usando minimização por quadrados mínimos, técnica descrita no algoritmo STAMP, na página 45.

Onde:

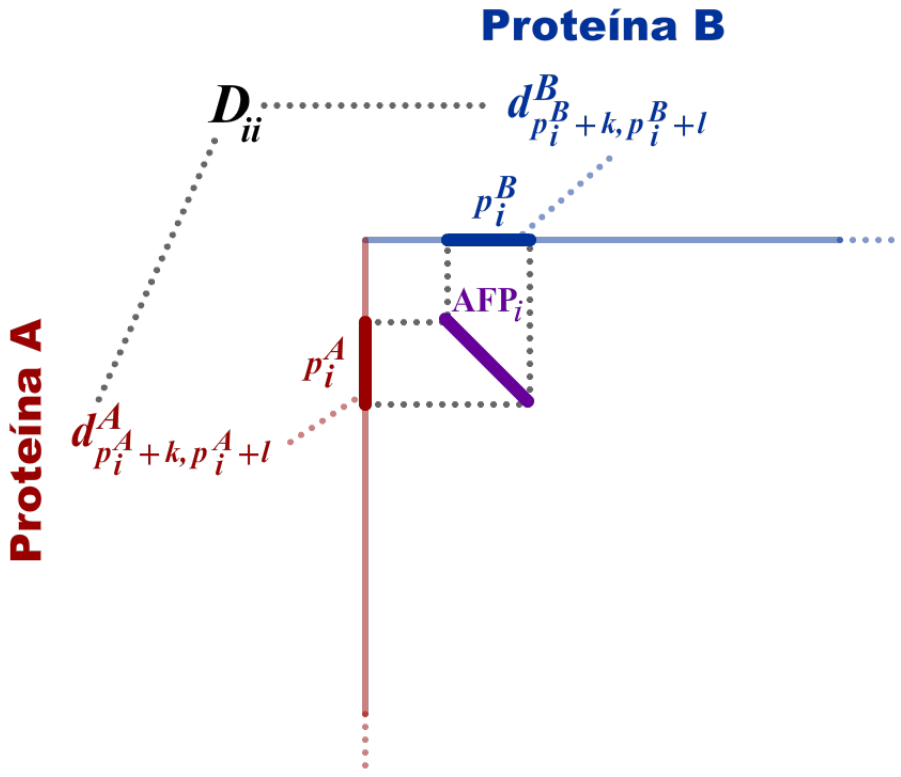
- D_{ij} denota a distância entre duas combinações de dois fragmentos das proteínas A e B, definida por dois AFPs nas posições i e j do caminho do alinhamento, onde $i \neq j$. No caso de um único AFP, ou seja, quando $i = j$, a distância é dada por D_{ii}. Um diagrama representativo das distâncias D_{ij} e D_{ii} é apresentado, respectivamente, nas Figuras 39 e 40;
- p_i^A denota a posição do resíduo inicial do AFP na proteína A na i-ésima posição do caminho do alinhamento;
- p_i^B denota a posição do resíduo inicial do AFP na proteína B na i-ésima posição do caminho do alinhamento;
- d_{ij}^A denota a distância entre os resíduos i e j na proteína A baseado na distância entre os átomos C^α;
- d_{ij}^B denota a distância entre os resíduos i e j na proteína B baseado na distância entre os átomos C^α;
- m denota o tamanho dos AFPs.

Figura 39 – Distância D_{ij} entre dois AFPs nas posições i e j onde $i \neq j$



Fonte: Adaptada de Shindyalov e Bourne (1998) pelo autor

Figura 40 – Distância D_{ii} de um único AFP na posição i onde $i = j$



Fonte: Adaptada de Shindyalov e Bourne (1998) pelo autor

A medida de distância (a) é usada para avaliar a combinação de dois AFPs, um como parte do alinhamento e o outro a ser adicionado. A medida (b) é usada para avaliar AFPs únicos, ou seja, o quão bem dois fragmentos que formam um AFP combinam. Por fim, a medida de distância (c) é utilizada para minimizar o tamanho do problema, selecionando os melhores alinhamentos e otimizando a quantidade de *gaps* no alinhamento final.

Para a extensão do caminho do alinhamento, podem ser usadas três estratégias no momento em que se vai inserir mais um AFP no caminho que está sendo calculado:

(i) Considerar todos os AFPs que podem estender o alinhamento e satisfazer os critérios de alinhamento;

(ii) Considerar apenas o AFP que pode estender de forma ótima o alinhamento além de satisfazer os critérios de alinhamento;

(iii) Usar uma estratégia intermediária.

Caso a primeira estratégia for escolhida, será realizada uma busca combinatória exaustiva pelo caminho ótimo, visto que todos os possíveis AFPs serão testados. Na segunda estratégia, o tamanho do conjunto de AFPs é mínimo, sendo assim, a quantidade de buscas por caminhos ótimos é menor que na primeira estratégia. Segundo Shindyalov e Bourne (1998), a utilização da segunda estratégia é suficiente para o trabalho de encontrar estruturas similares, além de, obviamente, ser menos custosa computacionalmente, visto que apenas um conjunto pequeno de AFPs será testado, ao passo que na primeira estratégia, todos os possíveis AFPs serão avaliados.

Ainda, existe a necessidade de se escolher os melhores pontos de início para o cálculo do caminho de alinhamento. De acordo com Shindyalov e Bourne (1998), essa escolha se dá primeiramente pelas regiões de maior nível de similaridade na matriz S, além de que, durante a busca do alinhamento de comprimento máximo, são descartados todos os alinhamentos de comprimentos que não estão alcançando o tamanho dos alinhamentos de tamanhos maiores.

Após a obtenção do maior caminho de alinhamento, este é avaliado de modo a verificar sua significância estatística. Esta significância é representada pelo *z-score*, avaliando para isso a probabilidade de se encontrar um caminho de alinhamento com uma quantidade menor ou igual de *gaps* que o caminho computado, além da distância da comparação de estruturas aleatórias utilizando um conjunto não-redundante, sendo que o alinhamento dessas estruturas aleatórias é feito da mesma forma que se faz para duas estruturas de interesse. O *z-score* é calculado numericamente ao se resolver

a Equação 2.27, tomando z como uma distribuição normal de média igual a 0 e desvio padrão igual a 1.

$$\rho(0_j 1, -z) = \rho(D_i^{av}, D_i^{sd}, D^{obs}) \cdot \rho(G_i^{av}, G_i^{sd}, G^{obs}) \tag{2.27}$$

Onde:

- z é o *z-score* do alinhamento, na forma:

$$\rho(\mu, \sigma, x) = \begin{cases} 2 \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}^e}^{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2} dy & \text{, se } x < \mu \\ 1 & \text{, caso contrário} \end{cases}$$

- i é o número do AFP no caminho de alinhamento;
- D^{obs} é o escore de distância observado (Equação 2.30);
- G^{obs} é a quantidade de *gaps* do caminho de alinhamento que está sendo calculado;
- D_i^{av} e D_i^{sd} são, respectivamente, a média e o desvio padrão da amostra, para escores de distância dos caminhos de tamanho i na comparação das estruturas aleatórias;
- G_i^{av} e G_i^{sd} são, respectivamente, a média e o desvio padrão da amostra, para escores dos *gaps* dos caminhos de tamanho i na comparação das estruturas aleatórias;

Para se decidir qual caminho deve ser estendido, as três heurísticas a seguir são utilizadas:

(i) Um único AFP (Equação 2.28);

(ii) AFPs em comparação com o caminho (Equação 2.29);

(iii) O caminho completo (Equação 2.30).

$$D_{nn} < D_0 \tag{2.28}$$

$$\frac{1}{n-1} \sum_{i=0}^{n-1} D_{in} < D_1 \tag{2.29}$$

$$\frac{1}{n^2} \sum_{i=0}^{n} \sum_{j=0}^{n} D_{ij} < D_1 \tag{2.30}$$

Onde:

- D_{ij} é a distância entre os AFPs i e j;
- n é o próximo AFP que será considerado para ser inserido no caminho de alinhamento $n-1$;
- D_0 e D_1 são limiares de similaridade com valores de 3Å e 4Å respectivamente.

Segundo Shindyalov e Bourne (1998), empiricamente, os alinhamentos obtidos que têm maior exatidão são calculados quando a escolha dos melhores AFPs e das extensões dos caminhos são feitas em três passos: (i) todos os AFPs candidatos são selecionados utilizando a Equação 2.28; (ii) o melhor AFP é escolhido utilizando a Equação 2.29; por fim, (iii) a decisão de estender ou finalizar o caminho é tomada utilizando a Equação 2.30.

O algoritmo de CE, desenvolvido por Shindyalov e Bourne (1998), foi utilizado por Shindyalov e Bourne (2001) na criação de um banco de dados de comparações estruturais de proteínas, além de ter sido a base para a criação de um servidor de comparação estrutural de múltiplas proteínas por Guda et al. (2004) e ter tido sua função de escore, usada na matriz de similaridade, aprimorada no trabalho de Jia et al. (2004). Atualmente, no endereço <http://cl.sdsc.edu/> da Web, pode ser encontrada a implementação atual do CE.

2.1.3.4 MATRAS

O *MArkov TRAnsition of protein Structure evolution* (MATRAS) é definido pelos seus autores, Kawabata e Nishikawa (2000), como um programa de computador desenvolvido de modo a ser capaz de comparar quaisquer pares de proteínas. O MATRAS utiliza três tipos de escores de similaridade que foram derivados do modelo de Markov, os quais serão detalhados adiante, e que tiveram como inspiração os escores de substituição de aminoácidos (*amino acid substitution score*) desenvolvidos por Dayhoff, Schwartz e Orcutt (1978) (matrizes PAM) e por Henikoff e Henikoff (1992) (matrizes BLOSUM), ambos já mencionados na seção "Comparação ou Alinhamento de Sequências". A forma geral dos escores de similaridade no MATRAS é dada, segundo Kawabata e Nishikawa (2000), pela Equação 2.31.

$$S(i,j) = \log \frac{P(j \rightarrow i)}{P(i)} \tag{2.31}$$

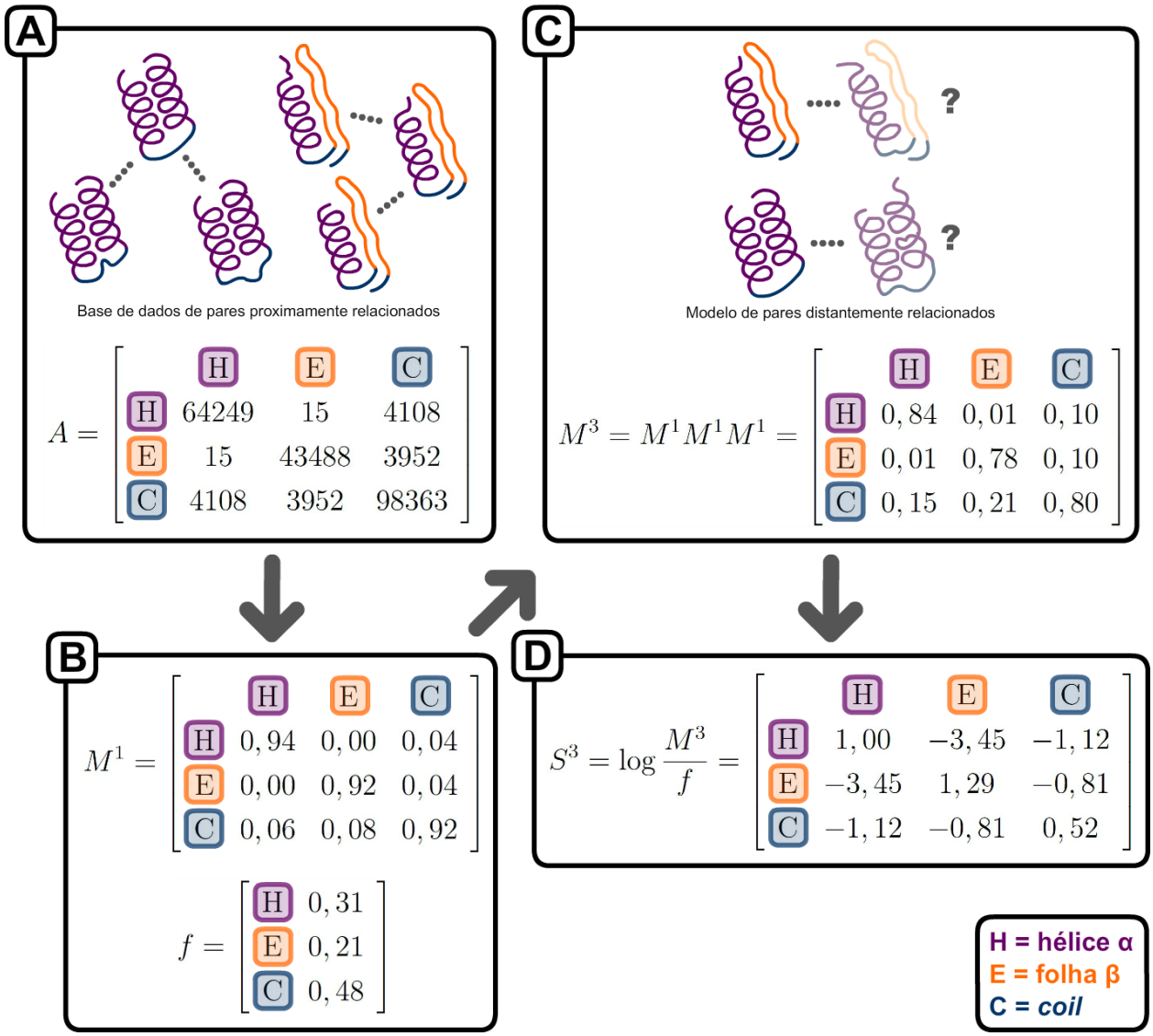
Onde:

- $P(j \rightarrow i)$ é a probabilidade da estrutura j se modificar ao ponto de se tornar a estrutura i;

- $P(i)$ é a probabilidade da estrutura i ser gerada ao acaso.

De acordo com Kawabata e Nishikawa (2000), para a obtenção dos escores de similaridade assume-se que a estrutura da proteína se modifica com o passar do tempo durante o processo de evolução, assim como a sequência de aminoácidos é alterada por meio de substituições nos modelos de Dayhoff, Schwartz e Orcutt (1978) e Henikoff e Henikoff (1992). A obtenção dos escores de similaridade foi feita, como já dito, utilizando o modelo de transição de Markov, sendo adaptado no chamado, segundo Kawabata e Nishikawa (2000), "Modelo Evolutivo de Transição Markoviano"[17], apresentado no esquema ilustrativo contido na Figura 41.

Figura 41 – Esquema do "Modelo Evolutivo de Transição Markoviano" utilizado no MATRAS



Fonte: Adaptada de Kawabata e Nishikawa (2000) pelo autor

[17] Tradução livre de *Markovian Transition Model of Evolution*.

Na seção A da Figura 41, é mostrado o primeiro passo do algoritmo do MATRAS, relativo ao agrupamento de proteínas de sequência similar. Após o agrupamento, são contadas as quantidades de transições (mudanças) estruturais, representadas na forma A_{ij}, onde i e j representam duas características estruturais, por exemplo, hélices α e folhas β na estrutura secundária. Nos diagramas da Figura 41, as colunas que representam estruturas do tipo hélice α estão coloridas em roxo, as que representam estruturas do tipo folha β estão coloridas em laranja e, por fim, as que representam *coils*, denominação usada pelos autores para descrever estruturas que não são do tipo das duas citadas anteriormente, estão coloridas em azul. Essa contagem é normalizada usando o peso $1/N_c$, onde N_c é a quantidade de proteínas em cada agrupamento.

Na seção B é representada a transformação da contagem de transições estruturais A_{ij} em uma probabilidade de transição, denotada por M^1_{ij}, em que o estado estrutural j se transforma no novo estado i. Essa transformação é feita de acordo com o modelo de Dayhoff, Schwartz e Orcutt (1978) utilizando as Equações 2.32 e 2.33.

$$M^1_{ij} = \frac{\lambda m_j A_{ij}}{\sum_{k \neq j} A_{kj}} \tag{2.32}$$

$$M^1_{jj} = 1 - \lambda m_j \tag{2.33}$$

Onde:

- λ é um parâmetro arbitrário;
- m_j é a medida de mutabilidade do estado j, dada por:

$$m_j = \frac{\sum_{k \neq j} A_{kj}}{\sum_k A_{kj}} \tag{2.34}$$

Além disso, as frequências dos tipos de subestruturas são também calculadas a partir dessas contagens.

A relação entre λ e ρ (razão da transição aceita ou, no modelo de Dayhoff, Schwartz e Orcutt (1978), porcentagem do PAM) é dada pela Equação 2.35:

$$\rho = 1 - \sum_i f_i M^1_{ii} = \lambda \sum_i f_i m_i \tag{2.35}$$

Onde:

- f_i é a frequência do estado i, e é dada pela Equação 2.36:

$$f_i = \frac{\sum_l A_{il}}{\sum_k \sum_l A_{kl}} \tag{2.36}$$

Quando se escolhe o valor de ρ, o parâmetro λ é calculado utilizando a Equação 2.35. Segundo Kawabata e Nishikawa (2000), o MATRAS utiliza o valor $\lambda = 1,00$, tornando o cálculo de M^1_{ij} (Equações 2.32 e 2.33 do modelo de Dayhoff, Schwartz e Orcutt (1978)) mais simples, visto que para $\lambda = 1,00$, a matriz M^1_{ij} será dada pela Equação 2.37:

$$M^1_{ij} = \frac{A_{ij}}{\sum_k A_{kj}} \tag{2.37}$$

Para se obter a matriz de probabilidades de mutação M^1 para escalas de tempo maiores, multiplica-se M^1 por ela mesma (seção C da Figura 41). A matriz de N passos evolutivos, M^N, é obtida dividindo a matriz de probabilidades de mutação por N, ou seja:

$$M^N = \frac{M^1}{N} \tag{2.38}$$

Como já apresentado, assume-se que as estruturas das proteínas sofrem mudanças com o passar do tempo, visto a influência do processo evolutivo. Usando a matriz de probabilidade M^N e as frequências f_i, o escore de similaridade definido na Equação 2.31 pode ser aproximado (seção D da Figura 41) usando a Equação 2.39:

$$S(i,j) = \log \frac{P(j \to i)}{P(i)} \simeq \log \frac{M^N_{ij}}{f_i} \tag{2.39}$$

Ainda, de acordo com Kawabata e Nishikawa (2000), na prática, para alguns casos do estado i, os valores obtidos ao se utilizar a Equação 2.37 são influenciados por pequenas mudanças na quantidade de observações, sendo assim, há a necessidade de se corrigir essa inconsistência, gerando para isso a matriz de probabilidade M'^N_{ij}, dada pela Equação 2.40:

$$M'^N_{ij} = \left(1 - \frac{m\sigma}{1 + m\sigma}\right) f_i + \frac{m\sigma}{1 + m\sigma} M^N_{ij} \tag{2.40}$$

Onde:

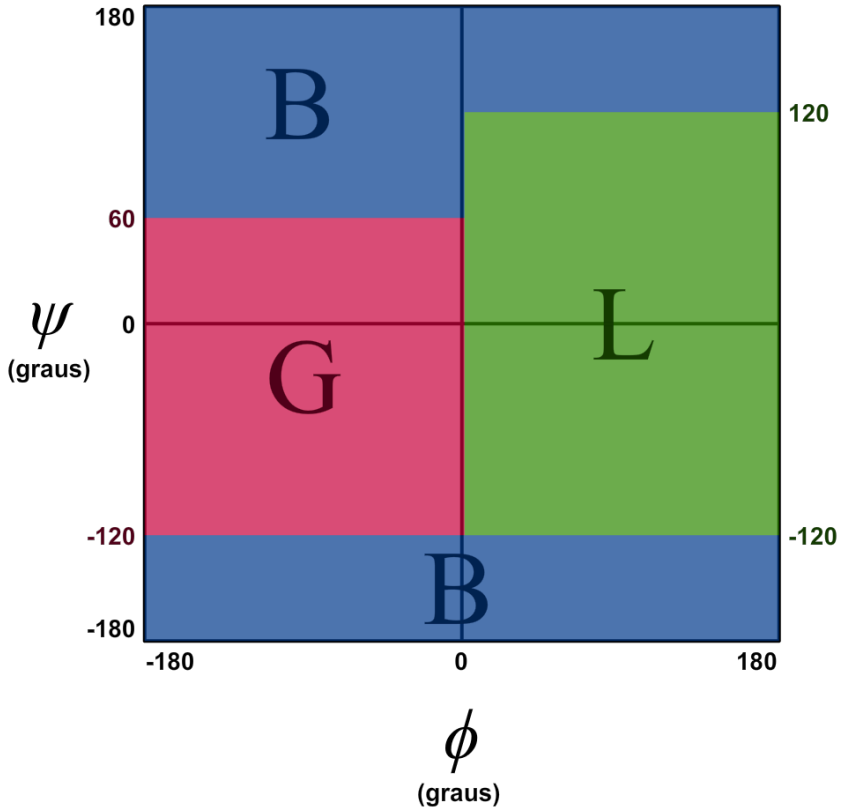
- σ é um parâmetro arbitrário, ajustado para 10.000,0 para o Escore Ambiental ou para 100,00 para o Escore de Elemento de Estrutura Secundária e para o Escore de Distância (detalhados a seguir);
- m é o valor mínimo ao se comparar as transições $j \to i$ e $i \to j$:

$$m = \min[M^N_{ij}, M^N_{ji}] \tag{2.41}$$

Pelo exposto, pode-se então definir, segundo Kawabata e Nishikawa (2000), os três tipos de escores de similaridade utilizados pelo MATRAS:

1. **S_{env} (Escore Ambiental):** Este escore mapeia o relacionamento entre o solvente e as estruturas da proteína. No MATRAS, as estruturas são classificadas em cinco tipos: hélice α (H), folha β (E) e *coils* com ângulos ϕ e ψ na região G ($-180 \leqslant \phi \leqslant 0 \wedge -120 \leqslant \psi \leqslant 60$), na região L ($0 < \phi \leqslant 180 \wedge -120 \leqslant \psi \leqslant 120$) e na região B (restante do plano ϕ-ψ). Um esquema ilustrativo das regiões G, L e B pode ser visto na Figura 42.

Figura 42 – Gráfico de Ramachandran para as regiões G, L e B



Fonte: Elaborada pelo autor

Em relação ao solvente, cada uma das cinco regiões apresentadas anteriormente podem estar expostas ou ocultas, sendo assim, há dez possíveis ambientes. O Escore Ambiental, S_{env}, é definido, segundo Kawabata e Nishikawa (2000), pela Equação 2.42:

$$S_{\text{env}}(R_i, R_j) = \log \frac{M^N(R_i, R_j)}{f(R_i)} \tag{2.42}$$

Onde:

- R_i é o i-ésimo resíduo de um ambiente de uma proteína;
- R_j é o j-ésimo resíduo de um ambiente de outra proteína.

Este escore é usado apenas para o cálculo do primeiro alinhamento provisório das duas proteínas, sendo necessário melhorar os resultados obtidos ao se utilizar os outros dois escores apresentados a seguir.

2. **S_{dis} (Escore de Distância):** Este escore calcula a distância entre os C^{β}[18] dos resíduos das duas proteínas que estão sendo comparadas. O cálculo do S_{dis} é dado pela Equação 2.43:

$$S_{\text{dis}}^{k}(D_{ix}, D_{jy}) = \log \frac{M_{k}^{N}(D_{ix}, D_{jy})}{f_{k}(D_{ix})} \tag{2.43}$$

Onde:

- D_{ix} é a distância entre o i-ésimo resíduo e o x-ésimo resíduo de uma proteína;
- D_{jy} é a distância entre o j-ésimo resíduo e o y-ésimo resíduo de outra proteína;
- k é a quantidade de resíduos que separa os resíduos i e x, sendo dada por $|i-x|$

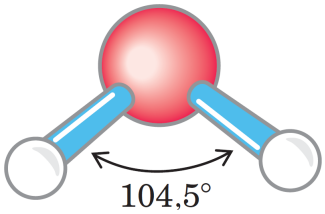
Separações curtas ($1 \leqslant k \leqslant 20$) são tratadas de forma independente, enquanto que separações longas ($k > 20$) são representadas em uma única matriz. Caso o alinhamento contenha *gaps*, k vai diferir entre as duas proteínas, visto que $|i-x| \neq |j-y|$ e nestes casos a maior diferença será utilizada, ou seja, $k = \max[|i-x|, |j-y|]$. Segundo Kawabata e Nishikawa (2000), o S_{dis} é utilizado apenas no estágio final de alinhamento do MATRAS, visto que ele é, dentre os três escores, o mais sensível na detecção de similaridades estruturais. Ainda, segundo Kawabata e Nishikawa (2000), escolheu-se utilizar a distância entre os C^{β}, ao invés da distância entre os C^{α}, por ser uma métrica que obtém melhores resultados para o alinhamento, especialmente em regiões formadas por conformações β.

3. **S_{sse} (Escore de Elemento de Estrutura Secundária):** Um Elemento de Estrutura Secundária, ou *Secondary Structure Element* (SSE), é definido por Kawabata e Nishikawa (2000) como um grupo contínuo de resíduos que formam uma hélice α ou uma conformação β. O arranjo geométrico espacial de um par de SSEs pode ser representado utilizando 6 parâmetros:

- L_1**:** a quantidade de resíduos da proteína 1;
- L_2**:** a quantidade de resíduos da proteína 2;
- θ_1**:** o ângulo de ligação (ângulo formado entre três átomos com pelo menos duas ligações, Figura 43) da proteína 1;
- θ_2**:** o ângulo de ligação da proteína 2;
- d**:** a menor distância entre o par de SSEs;
- ϕ**:** o ângulo diédrico entre o par de SSEs.

[18] O C^{β} é o primeiro carbono do grupo R, contado a partir do C^{α} da cadeia principal, de um resíduo de aminoácido (LESK, 2016).

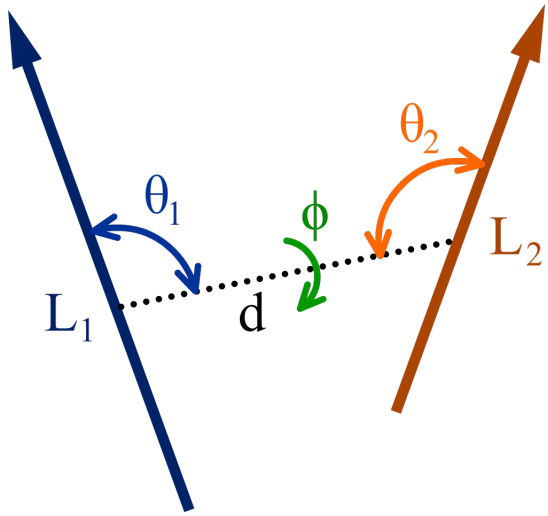
Figura 43 – Ângulo de ligação em uma molécula de água



Fonte: Adaptada de Nelson e Cox (2014) pelo autor

Estes parâmetros estão representados graficamente na Figura 44.

Figura 44 – Parâmetros que representam o arranjo geométrico espacial de um par de SSEs



Fonte: Adaptada de Kawabata e Nishikawa (2000) pelo autor

O cálculo do S_{sse} é dado pela Equação 2.44, sendo que os componentes desta Equação são definidos pelos sub-escores S^{θ}_{sse} (Equação 2.45), S^{ϕ}_{sse} (Equação 2.46), S^{d}_{sse} (Equação 2.47) e S^{L}_{sse} (Equação 2.48).

$$\begin{aligned} S_{\text{sse}}(i,x,j,y) = & S^{L}_{\text{sse}}(L_i, L_j) + S^{L}_{\text{sse}}(L_x, L_y) + \\ & S^{\theta}_{\text{sse}}(\theta_{1,ix}, \theta_{1,jy}) + S^{\theta}_{\text{sse}}(\theta_{2,ix}, \theta_{2,jy}) + \\ & S^{d}_{\text{sse}}(d_{ix}, d_{jy}) + S^{\phi}_{\text{sse}}(\phi_{ix}, \phi_{jy}) \end{aligned} \tag{2.44}$$

$$S^{\theta}_{\text{sse}}(\theta_{ix}, \theta_{jy}) = \log \frac{M^{N}(\theta_{ix}, \theta_{jy})}{f(\theta_{ix})} \tag{2.45}$$

$$S^{\phi}_{\text{sse}}(\phi_{ix}, \phi_{jy}) = \log \frac{M^{N}(\phi_{ix}, \phi_{jy})}{f(\phi_{ix})} \tag{2.46}$$

$$S^{d}_{\text{sse}}(d_{ix}, d_{jy}) = \log \frac{M^{N}(d_{ix}, d_{jy})}{f(d_{ix})} \tag{2.47}$$

$$S^L_{\text{sse}}(L_i, L_j) = \log \frac{M^N(L_i, L_j)}{f(L_i)} \tag{2.48}$$

Onde:

- i é o i-ésimo SSE da proteína 1;
- x é o x-ésimo SSE da proteína 1;
- j é o j-ésimo SSE da proteína 2;
- y é o y-ésimo SSE da proteína 2.

De acordo com Kawabata e Nishikawa (2000), este escore é utilizado no MATRAS para se obter os alinhamentos intermediários até que seja viável utilizar o escore de distância.

Sendo assim, o algoritmo de alinhamento implementado no MATRAS, segundo Kawabata e Nishikawa (2000), é composto de três estágios:

1. Utilização do S_{sse} para a construção iterativa dos N agrupamentos de SSEs, visando a melhoria incremental deste escore até que o valor obtido na iteração atual não seja melhor que o valor obtido na iteração anterior, obtendo assim o escore ótimo para os N agrupamentos obtidos;
2. Utilização de PD, com escore obtido usando o S_{env}, para a execução do algoritmo de alinhamento local;
3. A partir do alinhamento obtido no estágio anterior, novamente utiliza-se PD para melhorar o alinhamento, agora utilizando o S_{dis}, otimizando assim os resultados obtidos até se obter o alinhamento final. O cálculo do escore de similaridade para o i-ésimo resíduo de uma proteína com o j-ésimo resíduo da outra proteína é dado pela Equação 2.49.

$$S(i, j) = \frac{1}{2} \sum_a S^k_{\text{dis}}(D_{i,x(a)}, D_{j,y(a)}) \tag{2.49}$$

Onde:

- O par de resíduos $x(a)$ e $y(a)$ é obtido no alinhamento do estágio anterior.

Kawabata (2003) foi responsável em transformar o MATRAS em uma aplicação Web, além de tornar o algoritmo capaz de alinhar múltiplas estruturas ao invés de somente duas por vez, além de outras funcionalidades. A versão atual da ferramenta pode ser encontrada na Web a partir do endereço <http://strcomp.protein.osaka-u.ac.jp/matras/>.

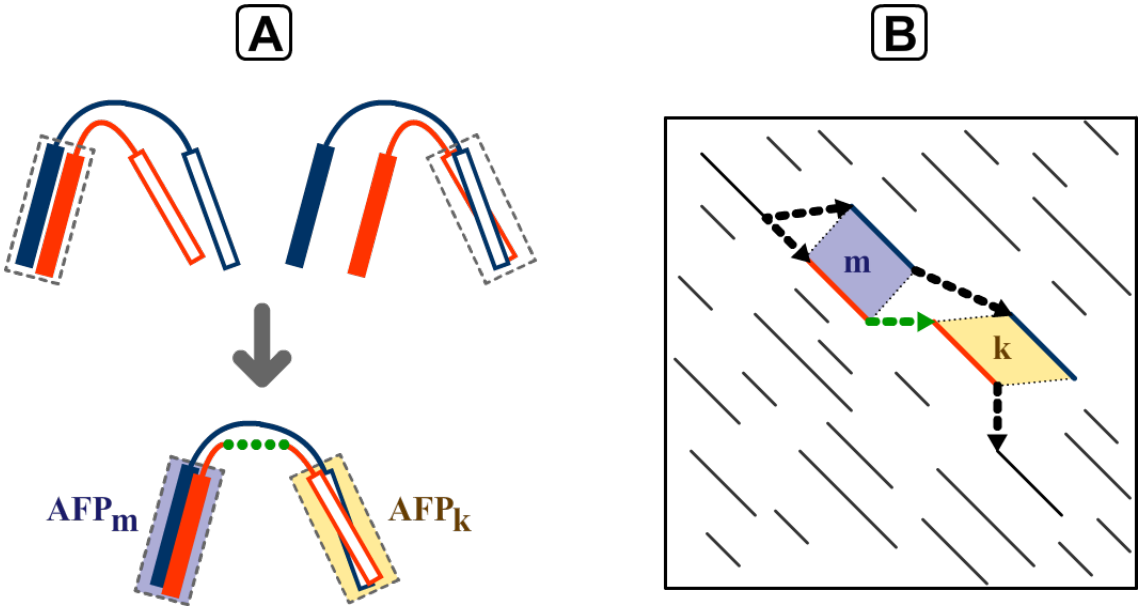
2.1.3.5 FATCAT

O *Flexible structure AlignmenT by Chaining AFPs with Twists* (FATCAT), desenvolvido por Ye e Godzik (2003), é um algoritmo de alinhamento de estruturas de proteínas que incorpora em seu funcionamento a funcionalidade de levar em consideração a flexibilidade conformacional das proteínas, ao passo que os outros algoritmos tratados até agora, tratam as proteínas como corpos rígidos. Nessa seção este algoritmo será definido com base no trabalho de Ye e Godzik (2003).

Dadas duas proteínas, A e B, quer-se obter o alinhamento estrutural ótimo das mesmas, com a menor quantidade possível de rearranjos, ou torções (*twists*), em uma das estruturas. Para isso, o FATCAT combina *gaps* e torções entre AFPs consecutivos para obter o alinhamento final.

Um AFP k, nas proteínas A e B, comparadas tem posições iniciais $b^A(k)$ e $b^B(k)$ e posições finais $e^A(k)$ e $e^B(k)$. Dois AFPs consecutivos são compatíveis quando superpõe as duas estruturas sem nenhuma modificação ou quando, para torná-los compatíveis, uma torção é inserida em uma das estruturas (seção A da Figura 45). No FATCAT, a posição relativa dos AFPs é sempre respeitada, ou seja, não haverá reposicionamento de AFPs para otimizar o alinhamento (seção B da Figura 45), sendo que para n torções, haverá $n + 1$ blocos de AFPs.

Figura 45 – Inserção de torção



Fonte: Adaptada de Ye e Godzik (2003) pelo autor

A detecção dos AFPs é feita de forma parecida com a do algoritmo CE de Shindyalov e Bourne (1998), ou seja, dois fragmentos de tamanho fixo L (8 resíduos) formam um AFP se o RMSD dos C^{α} desses fragmentos for menor que o limite de 3,0Å, limite este denominado de C_t.

A compatibilidade dos pares consecutivos de AFPs é calculada usando o RMSD entre as matrizes de distância dos resíduos que constituem os AFPs, sendo denotada por D_{mk} para os AFPs m e k (Equação 2.50). Uma alta similaridade (menor distância) implica em AFPs compatíveis. Caso a distância seja grande, insere-se as torções de modo a melhorar a similaridade (compatibilidade) entre os AFPs, como mostrado na seção A da Figura 45.

$$D_{mk} = \sqrt{\sum_{s=1}^{L} \left(d^{A}_{b^{A}(m)+s, b^{A}(k)+s} - d^{B}_{b^{B}(m)+s, b^{B}(k)+s} \right)^{2}} \tag{2.50}$$

Onde:

- $d^{A}_{i,j}$ é a distância entre os resíduos i e j na proteína A;
- $d^{B}_{i,j}$ é a distância entre os resíduos i e j na proteína B;
- $b^{A}(m)$ é a posição inicial do AFP m na proteína A;
- $b^{A}(k)$ é a posição inicial do AFP k na proteína A;
- $b^{B}(m)$ é a posição inicial do AFP m na proteína B;
- $b^{B}(k)$ é a posição inicial do AFP k na proteína B;
- L é o tamanho dos AFPs.

O alinhamento flexível entre duas estruturas é obtido a partir do encadeamento de AFPs com no máximo t torções, sendo que quando $t = 0$, o alinhamento flexível se torna um alinhamento rígido, visto que não há torções. O escore $S(k)$ (Equação 2.51) é utilizado para se decidir como o próximo AFP será conectado ao AFP k, obedecendo a algumas restrições apresentadas abaixo:

$$S(k) = a(k) + \max \left\{ \max_{\substack{e^{A}(m) < b^{A}(k) \\ e^{B}(m) < b^{B}(k)}} [(S(m) + c(m \rightarrow k)], 0 \right\} s \cdot t \cdot T(k) \leqslant t \tag{2.51}$$

Onde:

- $a(k)$ é o escore do AFP k, calculado utilizando a Equação 2.52;

$$a(k) = R_s \times L \times F(d_k) \tag{2.52}$$

Onde:

  - R_s é o escore de "recompensa", associado com um bom alinhamento (AFP longo e RMSD pequeno);
  - L é o comprimento do AFP;
  - $F(d_k)$ é uma função do RMSD (d_k) do AFP.

- $c(m \rightarrow k)$ é o escore da inserção da conexão entre os AFPs m e k (Equação 2.54);
- $T(k)$ é a quantidade de torções necessárias para se conectar os AFPs em $S(k)$, sendo calculado utilizando a Equação 2.53:

$$T(k) = T(m) + t(m \rightarrow k) \tag{2.53}$$

Onde:

  - $t(m \rightarrow k)$ é 1 caso seja preciso inserir uma torção para conectar os AFPs m e k e 0 caso contrário, ou seja, caso não haja necessidade de inserir uma torção para realizar a conexão entre m e k.

Por fim, o escore que define a conexão entre os AFPs m e k é dado pelas Equações 2.54, 2.55 e 2.56.

$$c(m \rightarrow k) = W(D_{mk}) \times P_c \times F(p, q) \tag{2.54}$$

$$W(D_{mk}) = \begin{cases} 1 & \text{, se } D_{mk} > D_c \\ \left(\frac{D_{mk} - D_0}{D_c - D_0}\right)^2 & \text{, senão se } D_0 < D_{mk} \leqslant D_c \\ 0 & \text{, caso contrário} \end{cases} \tag{2.55}$$

$$F(p, g) = M_c \times p + M_s \times q \tag{2.56}$$

Onde:

- $c(m \rightarrow k)$ é o escore de conexão entre os AFPs m e k;
- D_{mk} é o RMSD entre os AFPs m e k (já definido);
- D_c é o limiar para a definição da torção;
- D_0 é o limiar para penalização de uma conexão;
- P_c é a penalidade máxima para a conexão de dois AFPs;
- M_c é a penalidade associada ao não alinhamento (*mismatching*) de duas posições;

- M_g é a penalidade para um *gap*.
- p é a quantidade de regiões não alinhadas (*mismatched*);
- q é a quantidade de *gaps* para se conectar dois AFPs.

O FATCAT utiliza os parâmetros apresentados anteriormente com os seguintes valores: $t = 5$, $L = 8$, $C_t = 3,0$, $D_c = 5,0$, $D_0 = 1,0$, $R_s = 3,0$, $P_c = -25$, $M_c = -0,5$ e $M_g = -0,5$. O FATCAT foi originalmente implementado por Ye e Godzik (2003) utilizando a linguagem de programação C++, sendo compilado no sistema operacional Linux. Ye e Godzik (2004) desenvolveram um servidor para disponibilização na Web do serviço de alinhamento do FATCAT, que pode ser acessado pelo endereço <http://fatcat.burnham.org/>.

2.1.3.6 Outros Métodos de Comparação Estrutural

Além dos métodos apresentados anteriormente, existem também diversos outros métodos que propõe a solução aproximada do problema geral de alinhamento/comparação estrutural de proteínas, visto que esse problema é classificado computacionalmente como *Nondeterministic Polynomial time Hard* (NP-Hard)[19] como provado por Lathrop (1994). Na lista abaixo são apresentados, sucintamente, alguns desses métodos, organizados cronologicamente de acordo com o ano de publicação.

- **Da-Fu, Jiang e Zu-Kang (1994):** O algoritmo apresentado pelos autores utiliza informações de proteínas homólogas, ou seja, que têm similaridade evolutiva em suas estruturas primárias, para inferir a similaridade estrutural, bem como padrões de conformação que são utilizados em preditores estruturais para apoiar no processo de comparação;
- **Gibrat, Madej e Bryant (1996):** A ferramenta *Vector Alignment Search Tool* (VAST) foi desenvolvida por Gibrat, Madej e Bryant (1996) e consiste em uma ferramenta que apresenta resultados pré-calculados de regiões similares entre proteínas depositadas no PDB e no *Molecular Modeling Database* (MMDB), sendo que para realizar esses cálculos é utilizada a indução dos dados estruturais das proteínas em grafos. Esta ferramenta está disponível no *National Center for Biotechnology Information* (NCBI), a partir do endereço <http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml>, sendo que sua versão atual, o VAST+, foi desenvolvida por Madej et al. (2014) e pode ser acessada no endereço <http://structure.ncbi.nlm.nih.gov/Structure/vastplus/vastplus.cgi>;

[19] A classificação de complexidade computacional NP-Hard (NP-Difícil) compreende todos os problemas que não podem ser resolvidos em tempo polinomial conhecido, ou seja, a solução ótima para o problema não é possível de ser computada.

- **Suyama, Matsuo e Nishikawa (1997):** Nesse trabalho é apresentado um algoritmo de alinhamento de estruturas de proteínas que utiliza, além da PD, perfis de alinhamento 3D, que são matrizes de escores derivadas de subestruturas conhecidas, ou seja, várias estruturas de proteínas são processadas, gerando esses perfis, que por sua vez são utilizados para comparar outras proteínas, identificando seções das proteínas que casam com esses perfis, obtendo assim a similaridade estrutural;

- **Taylor (1999):** Nesse algoritmo é utilizada a matriz de distância das estruturas que estão sendo comparadas de modo a centralizar, iterativamente, seções dessas proteínas, sendo que os centros são escolhidos com base na similaridade dos resíduos centrais. Ao se realizar a centralização, verifica-se se os resíduos adjacentes aos centros têm estrutura geométrica similar, expandindo os alinhamentos quando essa comparação geométrica gera escores úteis, sendo que estes escores são obtidos também via PD. Este algoritmo, denominado *Structure Alignment Program* (SAP), é também descrito no trabalho de Taylor (2000);

- **Ortiz, Strauss e Olmea (2002):** A ideia do algoritmo *MAtching Molecular Models Obtained from THeory* (MAMMOTH), desenvolvido por Ortiz, Strauss e Olmea (2002), é permitir a comparação de estruturas de proteínas obtidas experimentalmente, com proteínas que tiveram suas estruturas geradas automaticamente por algum preditor estrutural, sem que tenha havido nenhum tipo de modificação manual nesta estrutura. A ferramenta Web que permite a utilização do algoritmo MAMMOTH pode ser encontrada no endereço <http://ub.cbm.uam.es/servers/mammoth/mammoth.php>;

- **Shatsky, Nussinov e Wolfson (2002):** O algoritmo FlexProt, desenvolvido por Shatsky, Nussinov e Wolfson (2002), permite a comparação de proteínas que têm estruturas rígidas, ou seja, que não têm variação conformacional, com proteínas que têm estruturas que permitem algum tipo de variação, chamadas de proteínas flexíveis, sendo que no endereço <http://bioinfo3d.cs.tau.ac.il/FlexProt/> pode-se encontrar a ferramenta de comparação apresentada;

- **Taylor (2002):** O autor apresenta um algoritmo que utiliza a teoria do grafos para realizar a comparação entre duas proteínas, induzindo dois grafos a partir das estruturas tridimensionais das mesmas e então, a partir desses grafos, realiza uma busca por subgrafos comuns, que por sua vez representam as subestruturas que são similares;

- **Ferrari, Guerra e Zanotti (2003):** Nesse trabalho é apresentada uma infraestrutura computacional em grade (*grid computing*[20]), bem como um algoritmo que faz

[20] A computação em grade pode ser definida como uma infraestrutura distribuída, capaz de realizar o processamento de quantidades massivas de dados, aproveitando o tempo ocioso dos processados das

uso do poder de processamento da mesma, para comparar estruturas de proteínas depositadas no PDB;

- **Kotlovyi, Nichols e Eyck (2003):** O algoritmo *Shape And Transformation* (SAT) utiliza uma abordagem geométrica para comparar as estruturas das proteínas, representando a cadeia de C^{α} de cada proteína como uma linha em um espaço tridimensional;

- **Sasin, Kurowski e Bujnicki (2003):** Os autores apresentam nesse trabalho uma ferramenta, denominada *STRUcture CLAssification* (STRUCLA), capaz de realizar o alinhamento de estruturas de múltiplas proteínas. A ferramenta pode ser encontrada na Web a partir do endereço <http://asia.genesilico.pl/strucla/>;

- **Shih e Hwang (2003):** Nesse trabalho é apresentado o algoritmo *Fast aLignment Algorithm for finding Structural Homology of proteins* (FLASH) que calcula a similaridade entre as subestruturas das proteínas comparadas utilizando a ideia de representá-las utilizando SSEs, abordagem também utilizada no algoritmo MATRAS de Kawabata e Nishikawa (2000);

- **Can e Wang (2004):** No trabalho em questão, o algoritmo apresentado utiliza também uma abordagem geométrica para comparação de curvas que representam as proteínas, assim como no trabalho de Kotlovyi, Nichols e Eyck (2003);

- **Comin, Guerra e Zanotti (2004):** É apresentado pelos autores um algoritmo, denominado de *PROtein STructure comparison* (PROuST), para comparação estrutural de proteínas que realiza a indexação das proteínas depositadas no PDB, utilizando para isso algumas características conformacionais das mesmas, e que faz uso dessa indexação para encontrar regiões similares entre duas ou mais proteínas. No trabalho de Ciriello, Comin e Guerra (2007) o algoritmo PROUST é melhorado, com o objetivo de acelerar o processamento dos alinhamentos estruturais, bem como utilizar, assim como no trabalho de Ferrari, Guerra e Zanotti (2003), uma infraestrutura em grade;

- **Shapiro e Brutlag (2004a):** Os autores apresentam um servidor/algoritmo, denominado FoldMiner, capaz de realizar buscas no PDB e inferir motivos nas estruturas pesquisadas de forma não supervisionada, bem como um algoritmo, o LOCK 2 desenvolvido por Shapiro e Brutlag (2004b), que também utiliza SSEs para realizar a comparação estrutural das proteínas pesquisadas através do FoldMiner;

- **Xu et al. (2004):** Nesse trabalho é apresentado o servidor Web ProteinDBS, que realiza indexação das proteínas do PDB utilizando a estrutura de dados *Entropy*

máquinas que fazem parte dessa grade.

Balanced Statistical (EBS) k-d *tree*, desenvolvida por Scott e Shyu (2003), e permite, a partir dessa indexação, a busca por similaridade entre estruturas de proteínas. A ferramenta pode ser acessada na Web a partir do endereço <http://proteindbs.rnet.missouri.edu/>;

- **Aghili, Agrawal e Abbadi (2005):** O algoritmo *Protein Alignment by Directional shape Signatures* (PADS) induz uma assinatura geométrica das conformações das subestruturas das proteínas depositadas no PDB e usa essa informação para realizar a comparação de estruturas de proteínas;

- **Chionh et al. (2005):** Os autores propõe o algoritmo *Structure-Conscious ALignment of secondary structure Elements* (SCALE) que utiliza SSEs para realizar a comparação estrutural de proteínas;

- **Pelta et al. (2005):** Nesse trabalho é utilizada a abordagem de realizar a comparação estrutural de proteínas a partir da superposição dos mapas de contato das mesmas. No trabalho de Pelta, González e Vega (2008) esse algoritmo é otimizado e são apresentados alguns experimentos para verificar sua acuracidade;

- **Yona e Kedem (2005):** O algoritmo apresentado nesse trabalho realiza diversas transformações, especificamente rotações, nos modelos tridimensionais de duas proteínas de modo a alinhá-las com base nas métricas de RMSD e *Unit-vector Root Mean Square Distance* (URMSD);

- **Zhang e Skolnick (2005):** O *Template/Model-Align* (TM-Align), algoritmo apresentado nesse trabalho, utiliza a métrica denominada *Template/Model-Score* (TM-Score) desenvolvida por Zhang e Skolnick (2004) e DP para realizar o alinhamento da estrutura de proteínas. Nos trabalhos de Sharma, Papanikolaou e Manolakos (2013) e de Sharma e Manolakos (2015) os autores utilizam o TM-Align de modo a adaptar a execução de algoritmos de comparação de proteínas para arquiteturas *multicore*[21]. A página Web <http://zhanglab.ccmb.med.umich.edu/TM-align/> pode ser utilizada para acessar a ferramenta *online*;

- **Aung e Tan (2006):** Os autores apresentam nesse trabalho o desenvolvimento do algoritmo MatAlign, capaz de alinhar estruturas de proteínas utilizando as matrizes de distância das mesmas;

- **Vesterstroem e Taylor (2006):** Nesse trabalho é apresentado o algoritmo *Flexible Alignment of Secondary structure Elements* (FASE), capaz de realizar alinhamentos não sequenciais em estruturas de proteínas, ou seja, alinhar uma subestrutura de

[21] Inserção de dois ou mais núcleos de processamento em um único circuito integrado, possibilitando a execução paralela de instruções, distribuindo a carga de execução de um determinado algoritmo em diversos núcleos.

uma proteína que se encontra na direção N-Terminal → C-Terminal, com uma subestrutura da outra proteína que está na direção contrária;

- **Zotenko, O'Leary e Przytycka (2006):** O método *Secondary Structure Element Footprint* (SSEF) desenvolvido pelos autores utiliza os SSEs das proteínas que estão sendo comparadas para obter um panorama geral da conformação das mesmas. No trabalho de Zotenko et al. (2007) os autores apresentam alguns experimentos, principalmente relacionados à velocidade de processamento do algoritmo;

- **Wu et al. (2007):** O algoritmo apresentado pelos autores utiliza, assim como outros algoritmos já apresentados, uma abordagem geométrica para realizar a comparação das subestruturas das proteínas comparadas;

- **Chu et al. (2008):** Nesse trabalho, os autores utilizam, além dos SSEs para iniciar o processamento da comparação das proteínas, técnicas de processamento de imagens para realizar os alinhamentos tridimensionais;

- **Csaba, Birzele e Zimmer (2008):** O *Phenotypic Plasticity Method* (PPM) desenvolvido pelos autores utiliza ideias parecidas com as do algoritmo MATRAS (KAWABATA; NISHIKAWA, 2000) procurando realizar as comparações com base na probabilidade de uma substrutura se tornar outra subestrutura, além do seu diferencial, que é levar em consideração a plasticidade fenotípica das proteínas analisadas;

- **Sippl e Wiederstein (2008):** Os autores apresentam o TopMatch, um conjunto de algoritmos que utilizam a abordagem de matrizes de distância para a comparação entre as estruturas das proteínas. No endereço <http://topmatch.services.came.sbg.ac.at/> da Web, o serviço do TopMatch pode ser acessado. Nos trabalhos de Sippl (2008) e Sippl et al. (2008) os autores apresentam várias considerações em relação ao TopMatch, principalmente em relação à forma de processar as características conformacionais das proteínas;

- **Eslahchi et al. (2009):** O algoritmo apresentado nesse trabalho utiliza uma abordagem geométrica para a superposição das subestruturas das proteínas comparadas de modo a minimizar o RMSD;

- **Malod-Dognin, Andonov e Yanev (2009):** Os autores propõe um algoritmo baseado em matrizes de distância e em encontrar cliques em grafos induzidos a partir das estruturas comparadas;

- **Guerler e Knapp (2010):** Nesse trabalho é apresentado um algoritmo de comparação de estruturas de proteínas não sequencial, ou seja, quando a comparação das subestruturas não obedece a ordem em que essas subestruturas estão distribuídas

em cada proteína, que utiliza internamente outros algoritmos apresentados neste trabalho, como o Dali (HOLM; SANDER, 1993), o CE (SHINDYALOV; BOURNE, 1998) e o TM-Align (ZHANG; SKOLNICK, 2005);

- **Zhang et al. (2010):** O algoritmo apresentado utiliza também uma abordagem geométrica para a comparação estrutural;

- **Galgonek, Hoksza e Skopal (2011):** Outro algoritmo que utiliza geometria computacional para o obtenção da comparação, sendo que, em contraste com os outros algoritmos apresentados que induzem curvas a partir do *backbone* das proteínas, esse algoritmo trata cada aminoácido como uma esfera, realizando o cálculo das similaridades utilizando informações de intersecção entre as mesmas;

- **Joseph, Srinivasan e Brevern (2011):** No algoritmo apresentado nesse trabalho, denominado *Improved Protein Block Alignment* (iPBA), a estrutura tridimensional das proteínas é convertida em um conjunto de *Protein Blocks* (PBs), que segundo Joseph, Srinivasan e Brevern (2011), são um conjunto de dezesseis pentapeptídeos que representam um alfabeto de conformações comuns às proteínas, simplificando um problema tridimensional para um problema unidimensional e então utiliza o algoritmo de alinhamento global de Needleman e Wunsch (1970) para alinhar as sequências de PBs. O iPBA é uma versão melhorada do algoritmo PBALIGN, desenvolvido por Tyagi et al. (2008). No trabalho de Gelly et al. (2011) o iPBA é utilizado na implementação de uma ferramenta *online* para alinhamento de estruturas de proteínas e que pode ser acessada pelo endereço <http://www.dsimb.inserm.fr/dsimb_tools/ipba/>. Ainda, no trabalho de Joseph, Srinivasan e Brevern (2012), o iPBA é usado como base para um algoritmo de alinhamento de múltiplas estruturas;

- **Kifer, Nussinov e Wolfson (2011):** Os autores apresentam o algoritmo *GlObal Structure SuperposItion of Proteins* (GOSSIP), que processa as proteínas de modo a gerar assinaturas nas regiões comparadas, ou seja, valores que representam determinado intervalo estrutural, permitindo então o alinhamento global das estruturas comparadas. A interface Web do GOSSIP pode ser acessada pelo endereço <http://bioinfo3d.cs.tau.ac.il/gossip/> da Web;

- **Li e Ng (2011):** Nesse trabalho é apresentado outro algoritmo que trabalha com a comparação de mapas de contato induzidos a partir da estrutura de cada proteína. Essa comparação se baseia em quais regiões dos mapas comparados se sobrepõe;

- **Liu, Srivastava e Zhang (2011):** O algoritmo apresentado pelos autores utiliza as informações geométricas, obtidas a partir das curvas das cadeias principais das proteínas comparadas, para realizar o processo de comparação estrutural;

- **Mernberger, Klebe e Hullermeier (2011):** O *Semiglobal Graph Alignment* (SEGA), algoritmo apresentado nesse trabalho, utiliza a teoria dos grafos para, a partir de grafos induzidos com base nas estruturas das proteínas, realizar o alinhamento de suas substruturas;

- **Razmara, Deris e Parvizpour (2012):** Nesse trabalho é apresentado o algoritmo denominado *Topology String Alignment Method for Intensive Rapid comparison of protein structures* (TS-AMIR), que realiza o alinhamento estrutural de proteínas em dois estágios: no primeiro estágio, as estruturas secundárias contidas em cada uma das duas estruturas comparadas são associadas umas às outras, gerando assim um mapa de correspondência, sendo que essa associação é obtida a partir da comparação da topologia dos fragmentos escolhidos; no segundo estágio, é executado um alinhamento resíduo a resíduo entre as estruturas associadas;

- **Ashby et al. (2013):** O algoritmo desenvolvido pelos autores, chamado de *efficient enumeration-based Protein structure Comparison* (ePC), é capaz de realizar o alinhamento estrutural de proteínas utilizando grafos e mapas de contato;

- **Jung et al. (2013):** Os ângulos de torção da cadeia principal das proteínas são utilizados, nesse trabalho, para realizar a comparação estrutural das proteínas;

- **Wang et al. (2013):** Nesse trabalho os autores apresentam o algoritmo DeepAlign, que realiza o alinhamento estrutural usando similaridade geométrica e informações evolutivas e funcionais das proteínas;

- **Iakovidou et al. (2014):** O algoritmo *DIStance and COsine measures* (DISCO) é capaz de realizar o alinhamento múltiplo de estruturas fazendo uso das distâncias calculadas entre os C^{α} dos resíduos, bem como os ângulos de torção contidos no *backbone* das estruturas comparadas;

- **Razmara, Parvizpour e Samira (2014):** No algoritmo apresentado nesse trabalho são utilizados SSEs e AFPs para realizar a comparação estrutural, além de ter a capacidade de realizar o alinhamento flexível entre as proteínas comparadas;

- **Terashi e Takeda-Shitaka (2015):** Nesse trabalho os autores apresentam o *Contact Area-Based Alignment* (CAB-align), um algoritmo de alinhamento flexível de estruturas de proteínas que realiza o alinhamento utilizando dados relativos aos contatos entre os resíduos das proteínas;

- **Zhao e Sacan (2015):** O UniAlign é outro algoritmo que utiliza a informação evolutiva obtida a partir da análise do alinhamento da estrutura primária das proteínas para realizar o alinhamento estrutural com o apoio da comparação de distância entre os resíduos;

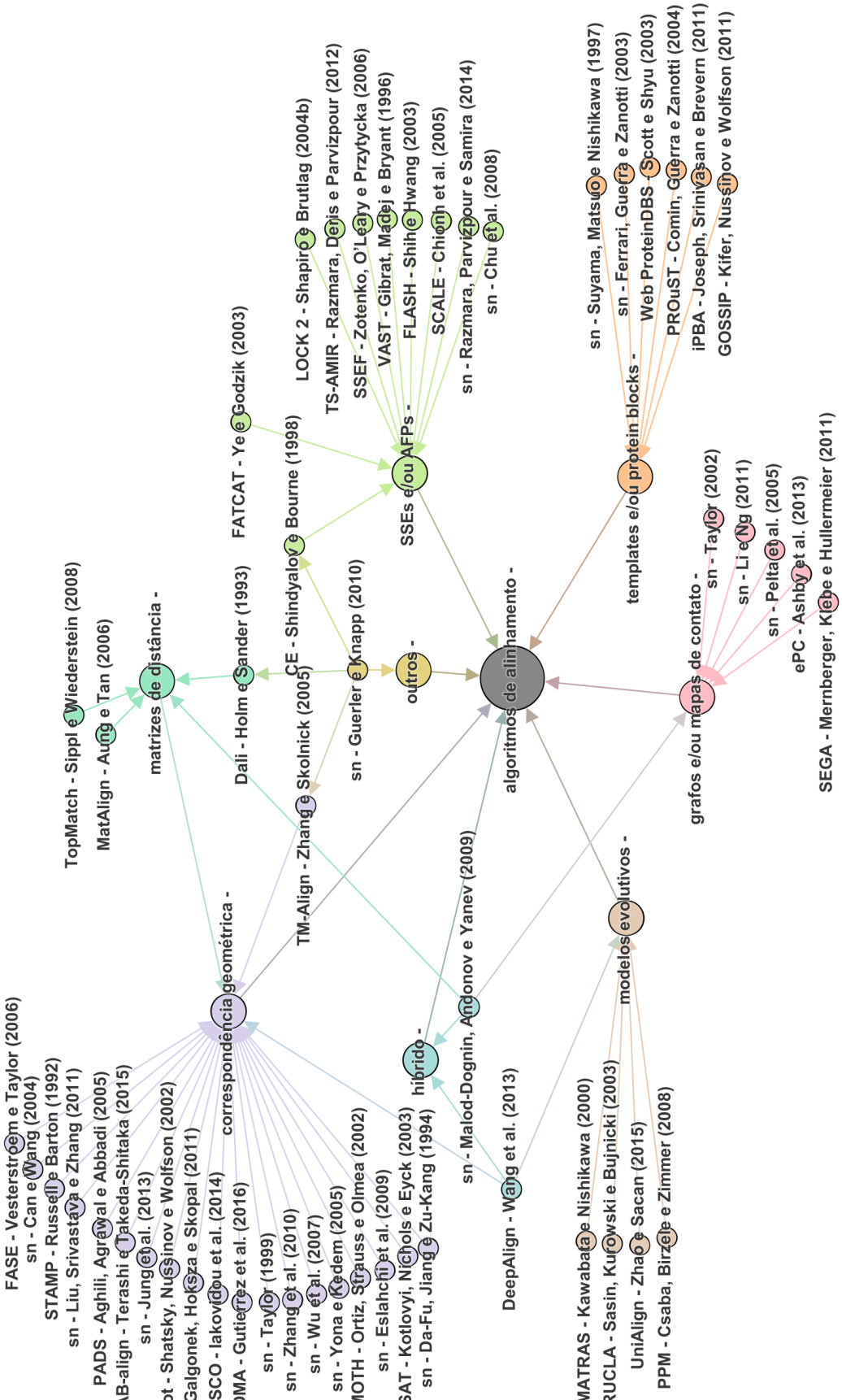
- **Gutierrez et al. (2016):** O algoritmo *MOrphing & MAtching* (MOMA), desenvolvido pelos autores, utiliza dados dos ângulos de torção do *backbone* das proteínas, bem como a distância entre as estruturas secundárias para realizar o alinhamento;

Apesar da lista de algoritmos de alinhamento estrutural ser extensa, existem ainda vários outros algoritmos que não foram citados, visto que, na maioria dos trabalhos relatados na mesma, é recorrente a informação de que os algoritmos Dali (HOLM; SANDER, 1993) e CE (SHINDYALOV; BOURNE, 1998) são os mais usados e com melhores resultados, sendo que esses dois algoritmos foram adotados, em conjunto com o FATCAT, na metodologia deste trabalho.

A seguir, na Figura 46, é mostrado um diagrama em forma de grafo que, baseado nas informações apresentadas pelos autores citados nessa seção, contém os algoritmos/trabalhos apresentados, classificados de modo a agrupar algoritmos que utilizam a mesma técnica básica para realizar o cômputo da similaridade estrutural entre as proteínas. O maior vértice, nomeado "algoritmos de alinhamento" é a origem direta ou indireta de todos os outros vértices, sendo que os de tamanho intermediário representam as técnicas utilizadas pelos algoritmos, que por sua vez, são os vértices menores, nomeados da forma "nome - trabalho". Os algoritmos que não tiveram seus respectivos nomes divulgados nos trabalhos estão nomeados no diagrama da Figura 46 como "sn", isto é, "sem nome".

Como exemplo, pode-se tomar o vértice do algoritmo CE, situado na parte superior central do diagrama, colorido em verde claro. Este algoritmo utiliza a técnica de "SSEs e/ou AFPs", sendo assim, ele está ligado com uma aresta direcionada ao vértice da técnica correspondente, que por sua vez, está ligada ao vértice central "algoritmos de alinhamento". Pode-se notar também que alguns algoritmos, classificados como "híbrido" (azul claro, esquerda do diagrama) utilizam mais de uma técnica. Por exemplo, o algoritmo DeepAlign, pois utiliza as técnicas de "correspondência geométrica" e "modelos evolutivos". Existe também no diagrama o trabalho de (GUERLER; KNAPP, 2010), que é um caso de um algoritmo que utiliza em seu funcionamento outros três algoritmos, sendo assim, ele está ligado nos mesmos (Dali, CE e TM-Align). Por fim, a técnica de "matrizes de distância" é um subtipo de "correspondência geométrica", estando então, ligada a esta técnica por uma aresta.

Figura 46 – Classificação dos algoritmos de alinhamento/comparação estrutural de proteínas apresentados



Fonte: Elaborada pelo autor

Pelo exposto, pode-se notar que existe uma infinidade de algoritmos de alinhamento estrutural de proteínas e que eles podem ser agrupados de acordo com as técnicas que são utilizadas para que o cômputo do alinhamento estrutural seja possível. Nessa seção foi apresentada uma das interfaces entre a Ciência da Computação e as Ciências Biológicas, que permite a geração de resultados, no caso, a comparação estrutural entre proteínas, algo de extrema importância. Nas próximas seções serão apresentadas as revisões bibliográficas sobre uma bactéria, o *Bacillus thuringiensis*, e um conjunto de proteínas que a mesma sintetiza, elementos fundamentais para o desenvolvimento deste trabalho.

2.2 *BACILLUS THURINGIENSIS*

O *Bacillus thuringiensis* (*Bt*) é uma bactéria aeróbia, Gram-positiva, encontrada no solo, diferenciando-se de outras bactérias do seu gênero (*Bacillus*) por sintetizar cristais proteicos intracelulares, produzidos durante a fase de esporulação, que são tóxicos a várias espécies de insetos, estes pertencentes às ordens *Lepidoptera*, *Diptera*, *Coleoptera*, *Himenoptera*, *Homoptera*, *Hemiptera*, *Isoptera*, *Mallophaga*, *Orthoptera*, *Phthraphera*, *Siphonaptera*, *Thisanoptera*, além de alguns ácaros, protozoários, nematoides e também a alguns tipos de células cancerígenas (ANGELO; VILAS-BÔAS; CASTRO-GÓMEZ, 2010; PINTO et al., 2009; PALMA et al., 2014).

Esses cristais são formados por diversos tipos de proteínas, sendo que seus principais constituintes, que são responsáveis pela toxicidade do *Bt* nos insetos das ordens mencionadas, são as proteínas Cristal, também chamadas de proteínas Cry, do inglês *Crystal Protein* (ANGELO; VILAS-BÔAS; CASTRO-GÓMEZ, 2010; PINTO et al., 2009). É importante destacar que além das proteínas Cry, o *Bt* é capaz de sintetizar vários outros tipos de proteínas úteis para os seres humanos. Na Tabela 4 são apresentados os principais tipos de proteínas sintetizadas pelo *Bt*, além de alguns detalhes pertinentes a cada um dos tipos, como variabilidade e natureza da atividade manifestada.

Tabela 4 – Diversidade de proteínas sintetizadas pelo *Bt*

Denominação	**Tipo da Proteína**	**Variabilidade**	**Atividade**
δ-endotoxinas Sintetizadas na Fase de Esporulação	Cry (*Crystal Protein*)	74 famílias ∼700 tipos	Inseticida Bactericida Antiparasitária Anticancerígena
	Cyt (*Cytolytic Protein*)	3 famílias ∼30 tipos	Inseticida (*Diptera* e *Coleoptera*)
Secreted Toxins Sintetizadas na Fase Vegetativa	Vip (*Vegetative Insecticidal Protein*)	4 famílias ∼150 tipos	Inseticida (*Coleoptera*)
	Sip (*Secreted Insecticidal Protein*)	1 tipo	Inseticida (*Coleoptera*)

Fonte: Adaptado de Palma et al. (2014) pelo autor

Em relação aos dados apresentados na Tabela 4, de acordo com Palma et al. (2014), pode-se verificar que existem diversas toxinas que são sintetizadas pelo *Bt*, sendo que as mesmas podem ser classificadas em quatro tipos divididos em duas denominações que, por sua vez, representam as duas diferentes fases do ciclo de vida do bacilo. As proteínas sintetizadas durante a fase de esporulação são comumente denominadas de δ-endotoxinas e compreendem as proteínas Cry, já mencionadas, e as proteínas Cyt. Durante a fase vegetativa do *Bt* outros dois tipos de proteínas são sintetizadas, as proteínas Vip e Sip, que são denominadas *Secreted Toxins*, ou seja, toxinas secretadas.

Segundo Roh et al. (2007), a primeira descrição do *Bt* foi feita no ano de 1915, quando Ernst Berliner o isolou a partir do inseto *Anagasta kuehniella*, uma traça da farinha. Esse isolamento aconteceu na província da Turíngia na Alemanha, dando assim o nome ao bacilo. Entretanto, ainda segundo Roh et al. (2007), o *Bt* foi descoberto inicialmente no Japão, por Ishiwata Shigetane, no ano de 1902, como uma bactéria contida no bicho-da-seda (*Bombyx mori*). Atualmente, sabe-se que a bactéria encontrada por Shigetane é uma subespécie do *Bt*. Por fim, a classificação do *Bt* é a seguinte: *Bacteria* (super-reino) > *Firmicutes* (filo) > *Bacilli* (classe) > *Bacillales* (ordem) > *Bacillaceae* (família) > *Bacillus* (gênero) > *Bacillus cereus* (grupo) > *Bacillus thuringiensis* (espécie)[22].

Na próxima Seção, as proteínas Cry, objeto de estudo deste trabalho, serão detalhadas, apresentando seu modo de ação e aplicações na agroindústria.

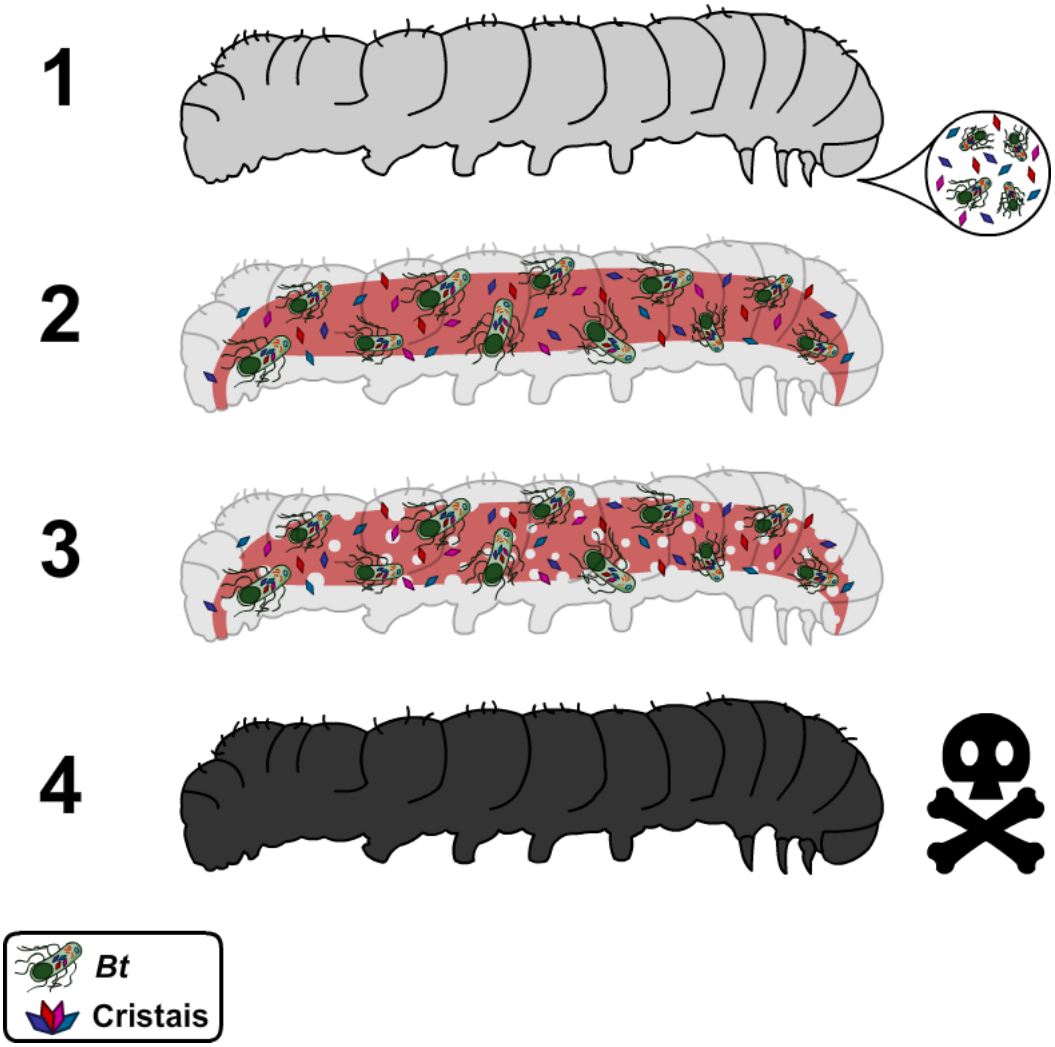
[22] A linhagem do *Bt* foi obtida através de consulta no site <http://www.ebi.ac.uk/>, do *European Bioinformatics Institute* (EBI), resultando nos dados apresentados no endereço <http://www.uniprot.org/taxonomy/1428>. Os dados da consulta ao *Toxonomy Browser* do NCBI para o Bt podem ser verificados no endereço <http://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?lvl=0&id=1428>

2.3 PROTEÍNAS CRY

As proteínas Cry, classificadas como toxinas formadoras de poros (*pore-formation toxins*) (BRAVO et al., 2004), possuem ação inseticida e são liberadas na forma de cristais na fase de esporulação do *Bt*. Esses cristais apresentam peso molecular entre 70 e 135 kDa (MIRANDA; ZAMUDIO; BRAVO, 2001) e tornam-se tóxicos às espécies sensíveis a eles quando são ingeridos. No intestino das larvas dos insetos afetados, as proteínas Cry são parcialmente digeridas e adquirem seu potencial tóxico, atuando na formação de poros nas paredes intestinais, fazendo com que as larvas morram de inanição e pela contaminação gerada pelas bactérias que existem no intestino dos insetos, gerando um quadro de septicemia (KNOWLES; DOW, 1993; PINTO et al., 2009). Na Figura 47 é apresentada uma ilustração do mecanismo de ação das proteínas Cry, dividido em quatro fases:

1. O inseto ingere cristais proteicos e esporos do *Bt* presentes no ambiente;
2. Os cristais são parcialmente digeridos, assumindo a conformação tóxica e se ligando à receptores específicos do intestino do inseto;
3. As proteínas criam poros na parede intestinal, contaminando o resto do organismo com esporos e outras bactérias presentes no intestino;
4. O inseto morre de inanição e pela infecção causada pelos esporos e pelas bactérias, que se proliferam no restante do corpo.

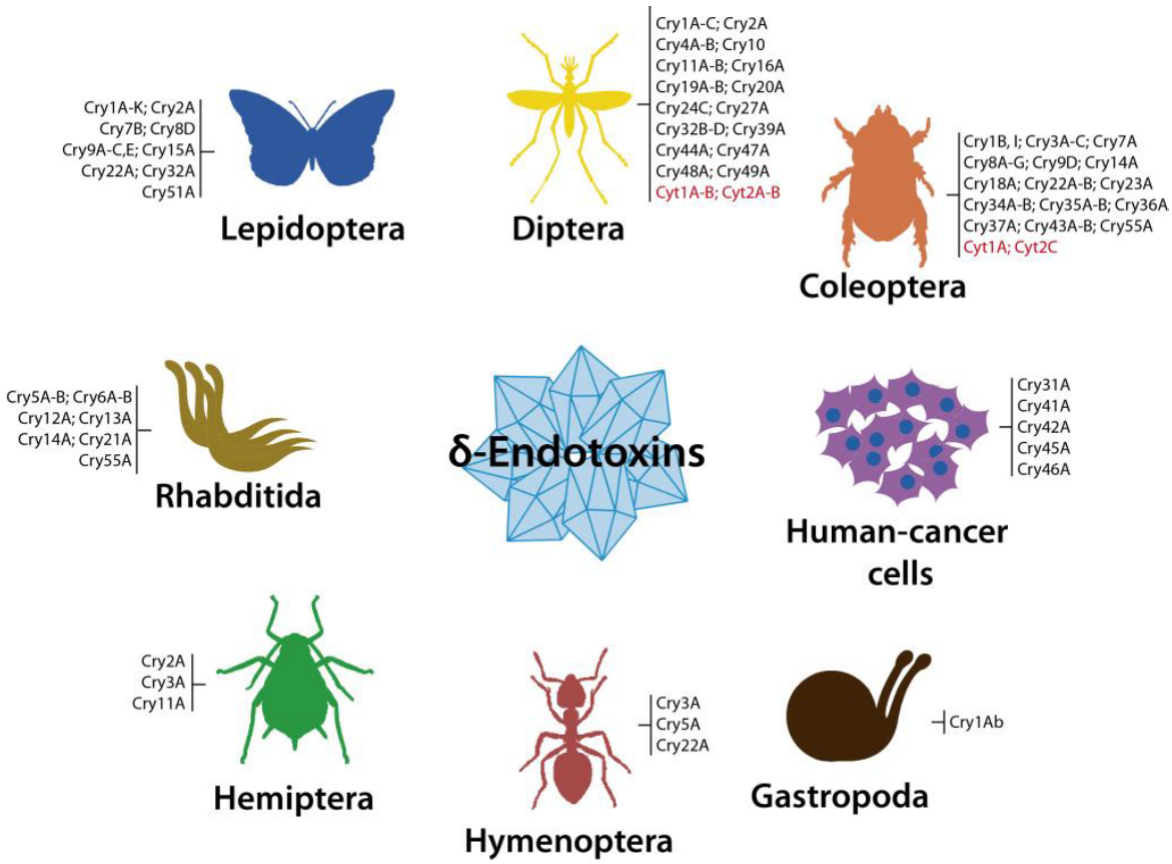
Figura 47 – Esquema representativo geral do modo de ação das proteínas Cry



Fonte: Adaptada de How... (2016) pelo autor

Existem diversas cepas de *Bt* que sintetizam diferentes tipos de proteína Cry e, como mencionado anteriormente, cada cepa, com suas respectivas proteínas, tem ação em diversas ordens de insetos, além de outros organismos e também em células cancerosas (PALMA et al., 2014). Na Figura 48 é apresentado um diagrama em que as diversas famílias das proteínas Cry são relacionadas com as ordens que cada uma afeta.

Figura 48 – Proteínas Cry e Cyt relacionadas com os alvos afetados



Fonte: Reproduzida na íntegra de Palma et al. (2014) pelo autor

Pode-se notar que, a partir dos dados apresentados na Figura 48, existem proteínas Cry que também são ativas em mais de uma ordem de insetos. Por exemplo, a proteína Cry1A é ativa contra as ordens *Lepidoptera* e *Diptera*, enquanto a proteína Cry1B é ativa contra as ordens *Lepidoptera*, *Diptera* e *Coleoptera*. Com isso, é possível inferir que provavelmente devem existir receptores comuns entre essas ordens, ou que talvez existam receptores diferentes em cada ordem, mas que viabilizam a manifestação tóxica das mesmas famílias de proteínas Cry. Diante da variabilidade de proteínas Cry existentes, foi definido um sistema de classificação para organizá-las, sendo que o mesmo será apresentado na próxima Seção.

2.3.1 Classificação

A classificação atual das proteínas Cry foi proposta por Crickmore et al. (1998) e tem como base o alinhamento dos aminoácidos das proteínas. Segundo Crickmore et al. (1998), o primeiro passo para a criação da nomenclatura, e que é utilizado atualmente para a classificação de novas proteínas, consiste em submeter as sequências dos aminoácidos de todas as proteínas ao algoritmo ClustalW, desenvolvido por Thompson, Higgins e Gibson (1994), que alinha as sequências par a par, gerando uma matriz de distância, que

quantifica as similaridades entre as proteínas. Após a obtenção desses dados é gerada uma árvore filogenética, que é uma representação gráfica do relacionamento evolutivo entre as proteínas submetidas ao algoritmo ClustalW.

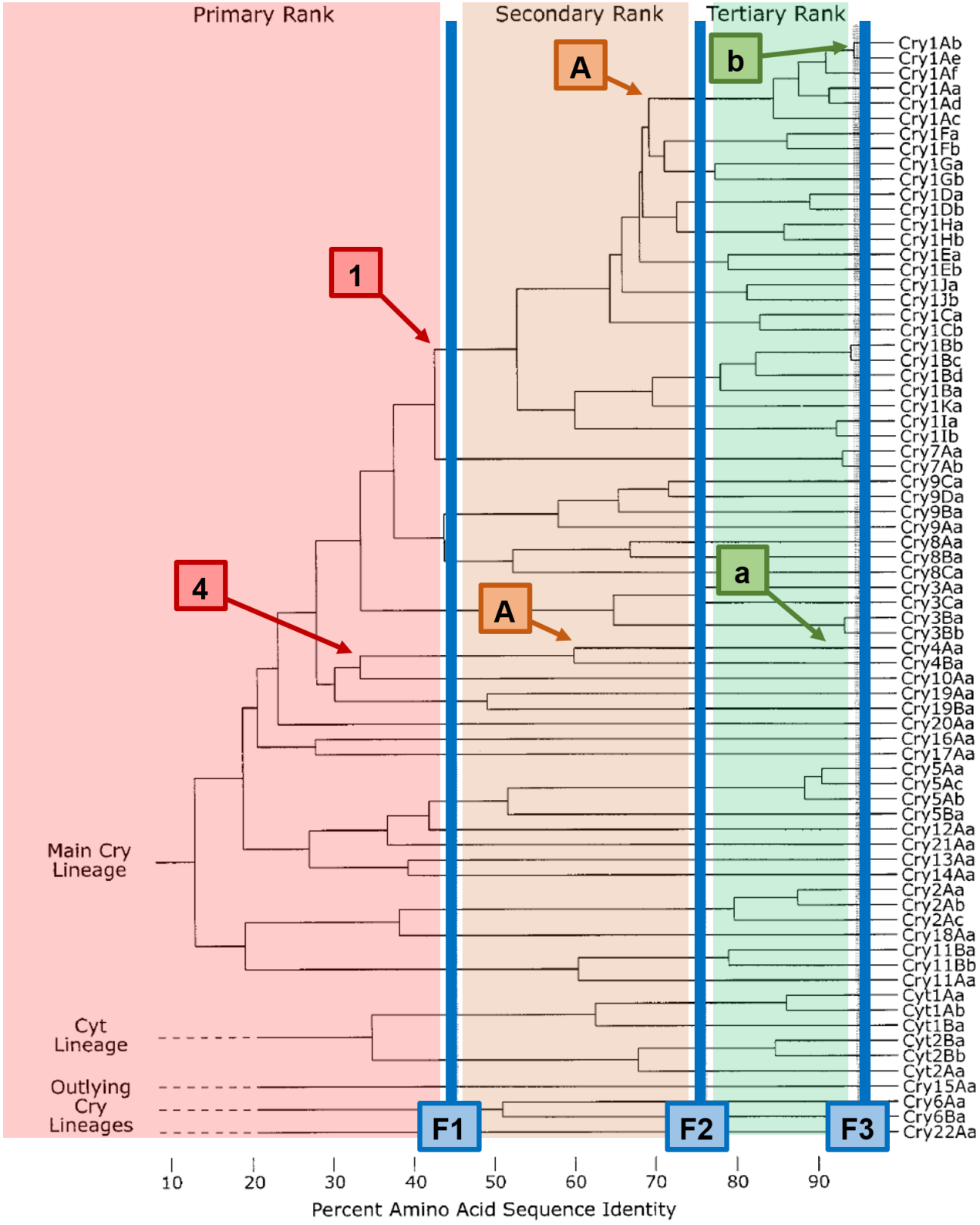
Com a árvore gerada, esta é dividida em três seções, apresentadas em vermelho (*primary rank*), laranja (*secondary rank*) e verde (*tertiaty rank*) na Figura 49, sendo essas utilizadas como intervalos para o ranqueamento usado na nomenclatura. Esses intervalos são limitados por fronteiras, também apresentadas na Figura 49. O nome dado a cada toxina depende da localização que cada proteína está situada em relação a essas fronteiras.

A toxina que entrar na árvore antes da fronteira "F1", será associada a um ranque primário utilizando um número arábico. A toxina que entrar entre as fronteiras "F1" e "F2", será associada a um ranque secundário utilizando uma letra maiúscula do alfabeto latino. A toxina que entrar entre as fronteiras "F2" e "F3", será associada a um ranque terciário utilizando uma letra minúscula do alfabeto latino e, por fim, toxinas que possuem sequências idênticas, mas que foram isoladas de forma independente, receberão um ranque quaternário, que é representado por outro número arábico.

Por exemplo, a proteína Cry1Ab apresentada na Figura 49 tem ranque primário igual a "1", ranque secundário igual a "A" e ranque terciário igual a "b". Ainda, na Figura 49, pode-se verificar o caminho seguido até chegar na classificação desta proteína. Outro exemplo, da proteína Cry4Aa, pode ser visto também na Figura 49.

Analisando as fronteiras apresentadas na Figura 49, pode-se verificar que as proteínas de ranques primários "1" e "7" têm cerca de 45% homologia em suas sequências. Outras verificações podem ser feitas ao se observar a porcentagem de aminoácidos que apresentam identidade nas sequências, pois esses valores estão inseridos no eixo "*Percent Amino Acid Sequence Identify*" da Figura 49.

Figura 49 – Exemplo de árvore filogenética de um conjunto de proteínas Cry



Fonte: Adaptada de Crickmore et al. (1998) pelo autor

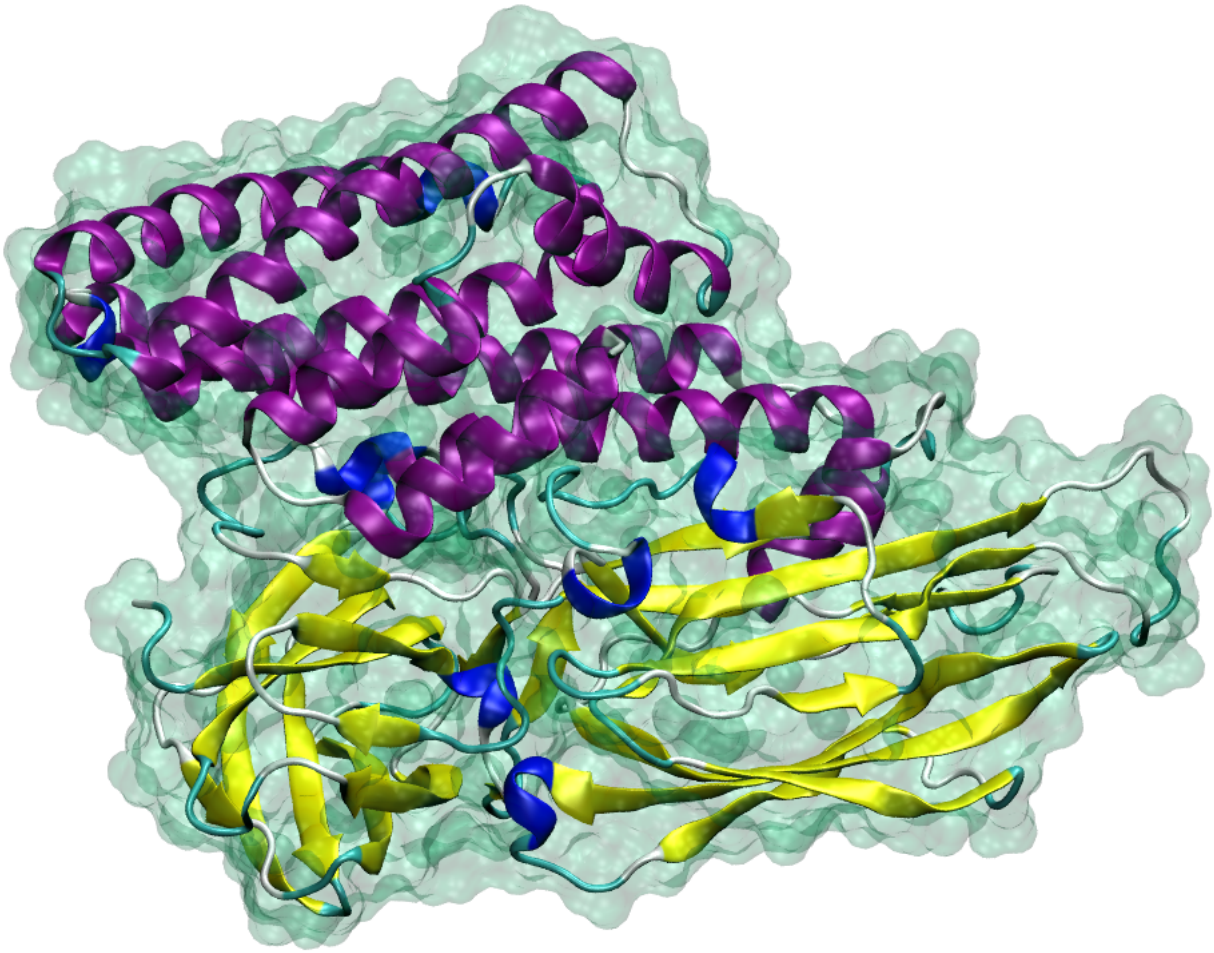
A maior parte das proteínas Cry apresentam estrutura tridimensional similar, sendo que esta estrutura está dividida, na maioria das vezes, em três domínios proteicos (MAAGD; BRAVO; CRICKMORE, 2001). Na próxima Seção serão apresentadas as características estruturais das proteínas Cry.

2.3.2 Caracterização Estrutural

Segundo Maagd, Bravo e Crickmore (2001), a estrutura das proteínas Cry normalmente é dividida em três domínios, sendo que o Domínio I, n-terminal, possui sete hélices α e tem participação na inserção na membrana intestinal do inseto e na formação do poro. O Domínio II, chamado de prisma β, apresenta três folhas β dobradas simétricas e, por fim, o Domínio III, c-terminal, é formado por duas folhas β antiparalelas. Ainda, de acordo com Maagd, Bravo e Crickmore (2001), os Domínios II e III estão envolvidos no reconhecimento dos receptores e na ligação da proteína na parede intestinal, além de o Domínio III também estar envolvido na formação dos poros.

A estrutura da proteína Cry1Aa1 (PDB: 1CIY), em sua conformação tóxica, obtida por Knowles e Ellar (1987) e Grochulski et al. (1995), pode ser vista na Figura 50, sendo que a estrutura da proteína está colorida de acordo com os tipos de estruturas secundárias (roxo para hélices α, amarelo para as conformações β, azul para outros tipos de hélice e branco/verde para *coils*), além de sua superfície aproximada estar representada em verde.

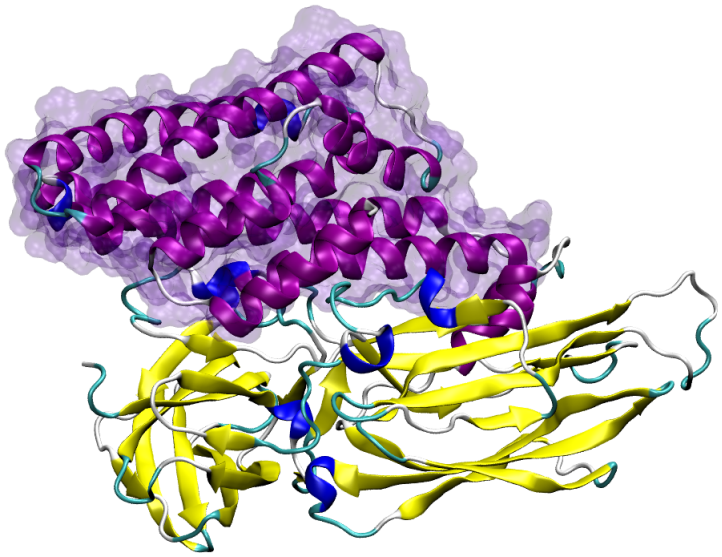
Figura 50 – Estrutura completa da conformação tóxica da proteína Cry1Aa1 (PDB: 1CIY)



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

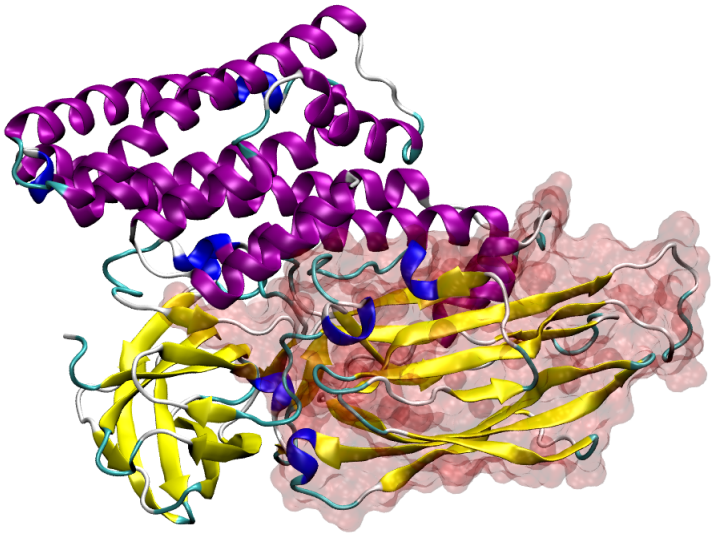
A estrutura apresentada na Figura 50 pode ser dividida nos três domínios citados anteriormente, sendo que os destaques desses domínios podem ser vistos, respectivamente, nas Figuras 51, 52 e 53.

Figura 51 – Destaque em roxo da superfície do Domínio I da conformação tóxica da proteína Cry1Aa1 (PDB: 1CIY)



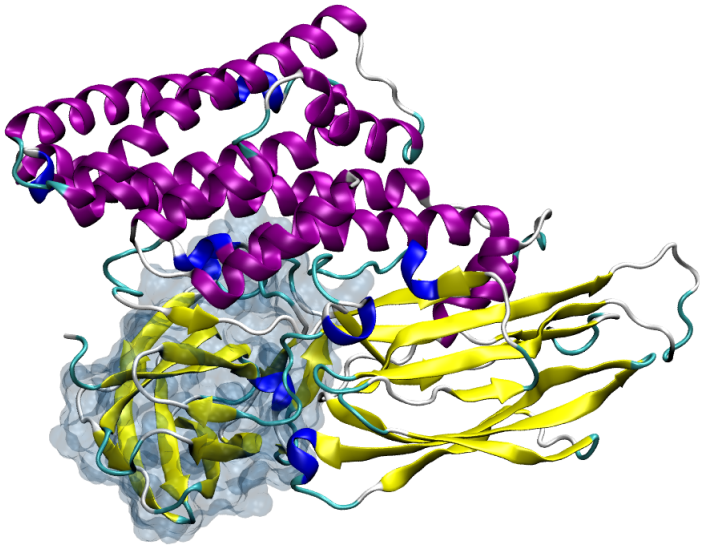
Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

Figura 52 – Destaque em vermelho da superfície do Domínio II da conformação tóxica da proteína Cry1Aa1 (PDB: 1CIY)



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

Figura 53 – Destaque em azul da superfície do Domínio III da conformação tóxica da proteína Cry1Aa1 (PDB: 1CIY)



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

Pelo exposto, cada um dos domínios das proteínas Cry têm importância na interação com os receptores das mesmas. Sendo assim, na próxima Seção serão apresentados os detalhes relacionados ao modo de ação das proteínas Cry em algumas ordens de insetos, com foco na atividade dos diferentes receptores envolvidos no processo.

2.3.3 Interação com Receptores

De forma mais detalhada, segundo Knowles e Dow (1993), a atividade das proteínas Cry é desencadeada a partir da solubilização, em ambientes de pH alto (acima de 9,5), e pela posterior digestão parcial das mesmas, executada a partir de proteases específicas do intestino dos insetos, ativando assim o potencial tóxico das proteínas. A clivagem das protoxinas, isto é, das proteínas Cry antes de adquirirem sua conformação tóxica, segundo Miranda, Zamudio e Bravo (2001), é feita inicialmente a partir da extremidade c-terminal e posteriormente na extremidade n-terminal, gerando, ao fim, as toxinas ativas, que tem peso molecular entre 60 e 65 kDa (BRAVO et al., 2004).

Ao assumirem seu estado ativo, de acordo com Knowles e Dow (1993), as toxinas se ligam aos receptores contidos em vesículas da *Brush Border Membrane* (BBM) das células colunares do intestino do inseto e, posteriormente, inserem-se de forma irreversível na membrana plasmática da célula, iniciando assim a formação de um poro, ou lesão na membrana, levando a uma quebra da permeabilidade, que por sua vez acarreta na lise das células afetadas e no rompimento da integridade intestinal do organismo, fazendo com que o mesmo morra por inanição e/ou por septicemia.

Em relação à ordem *Lepidoptera*, Bravo, Gill e Soberon (2007) afirmam que existem pelo menos quatro tipos de receptores que estão envolvidos no processo de ligação das proteínas às vesículas da BBM:

1. Uma proteína da família das caderinas (CADR), que segundo Bretschneider, Heckel e Pauchet (2016), é denominada HevCaLP em *Heliothis virescens*, Bt-R_1 (identificada por Vadlamudi, Ji e Bulla L. A. (1993) e por Martinez-Ramirez et al. (1994)) em *Manduca sexta* e Bt-R_{175} em *Bombyx mori*;
2. Uma aminopeptidase-N (APN) ancorada a uma âncora glicosilfosfatidilinositol (GPI);
3. Uma fosfatase alcalina (ALP) ancorada a uma âncora GPI;
4. Um glicoconjugado de 270kDa.

De acordo com Buss e Callaghan (2008) há um receptor da família dos transportadores *ATP-Binding Cassette* (ABC), denominado HevABCC2, que é fundamental no processo de infecção pelas proteínas da família Cry1A.

Apesar de existirem proteínas Cry específicas a determinadas ordens e outras que são tóxicas a mais de uma ordem, o mecanismo que desencadeia essa especificidade ainda não foi totalmente elucidado, entretanto é sabido que alterações na estrutura primária das toxinas, ou seja, trocas de aminoácidos em determinadas posições, são capazes de inibir o potencial tóxico das mesmas. Tiewsiri e Angsuthanasombat (2007) conduziram um experimento que constituía na modificação do gene da proteína Cry4Ba, específica à ordem *Diptera*, de modo a trocar quatro resíduos aromáticos altamente conservados, $^{242}W^{244}$, $^{245}F^{247}$, $^{248}Y^{250}$ e $^{263}F^{265}$, por uma Alanina (A), gerando proteínas modificadas que foram menos tóxicas ao mosquito *Stegomyia aegypti* do que a forma não modificada das mesmas.

Diversos trabalhos têm sido desenvolvidos com o objetivo de desvendar qual é o processo de ligação das proteínas Cry com os receptores citados anteriormente:

- Francis e Bulla (1997) estabeleceram a relação do receptor Bt-R_1 com a toxicidade ao tipo Cry1Ac;
- Keeton e Bulla (1997) determinaram a especificidade do Bt-R_1 com os tipos Cry1Aa, Cry1Ab e Cry1Ac no mandarová-do-fumo (*Manduca sexta*);
- Miranda, Zamudio e Bravo (2001) descobriram que a hélice α_1 do Domínio I da proteína Cry1Ab não é essencial para sua atividade;
- Gomez et al. (2002) relacionaram o intervalo [869; 876] (HITDTNNK) de aminoácidos do Bt-R_1 com a volta 2 do Domínio II da proteína Cry1A, compreendida no intervalo [363; 373] (SSTLYRRPFNI);

- Gomez et al. (2002) determinaram que o processo de ligação da proteína Cry1A com o receptor Bt-R_1 facilita a clivagem proteolítica da hélice α_1 do Domínio I, acelerando a formação de um tetrâmero, composto por quatro unidades de Cry1A, responsável pela inserção na membrana plasmática através de uma interação com o receptor APN;

- Bravo et al. (2004) estabeleceram que a oligomerização em um tetrâmero dispara a ligação da proteína Cry1Ab na APN, além de que a toxicidade não é apenas manisfestada no receptor Bt-R_1, visto que células que não estão presentes no intestino de *Manduca sexta* também possuem esse receptor, mas precisam de uma concentração muito alta de proteínas tóxicas para que sejam infectadas;

- De acordo com Zhang et al. (2005) o tipo Cry1Ab se incorpora na membrana celular tanto na forma monomérica quanto na forma oligomérica, sendo que a forma monomérica liga-se especificamente ao receptor Bt-R_1 e a forma oligomérica é lipídico-dependente e não específica, além de não ser responsável pela formação de poros e por consequência não matam as células do inseto. Os autores ainda apresentam que a morte da célula está relacionada à forma monomérica da toxina;

- Gomez et al. (2006) confirmam que os receptores Bt-R_1 e APN são receptores de Cry1A e que Cry1Ab liga-se ao Bt-R_1 promovendo a formação de um oligômero pré-poro que por sua vez se liga a APN promovendo a inserção na membrana. Além disso, afirmam que as voltas 2 e 3 do Domínio II da Cry1Ab são as que interagem inicialmente com o Bt-R_1 para a formação do oligômero e que a conformação β_{16} do Domínio III interage com a APN para a posterior inserção do oligômero na membrana celular;

- Gomez et al. (2007) afirmam que as proteínas Cry1A, que são tóxicas contra a ordem *Diptera*, têm efeito sinergético nesses organismos com a proteína Cyt1Aa, também sintetizada pelo *Bt*, pois essa liga-se a um receptor do intestino do inseto e então atua como receptor da proteína Cry. Além disso, confirmam também que existem três regiões da CADR que interagem com três voltas do Domínio II da proteína Cry1A, sendo elas:

  1. A volta 2 interage com os resíduos ^{865}NITIHITDTNN^{875} da CADR localizados na repetição 7;
  2. As voltas α 8 e 2 interagem com os resíduos ^{1331}IPLPASILTVTV^{1342} da CADR localizados na repetição 11;
  3. Um terceiro trecho da CADR localizado na repetição 12. No caso da CADR do organismo *Heliothis virescens*, outro tipo de mandarová-do-fumo, a volta 3 do Domínio II da proteína Cry1Ac interage com os resíduos ^{1423}GVLTLNFQ^{1431} da caderina.

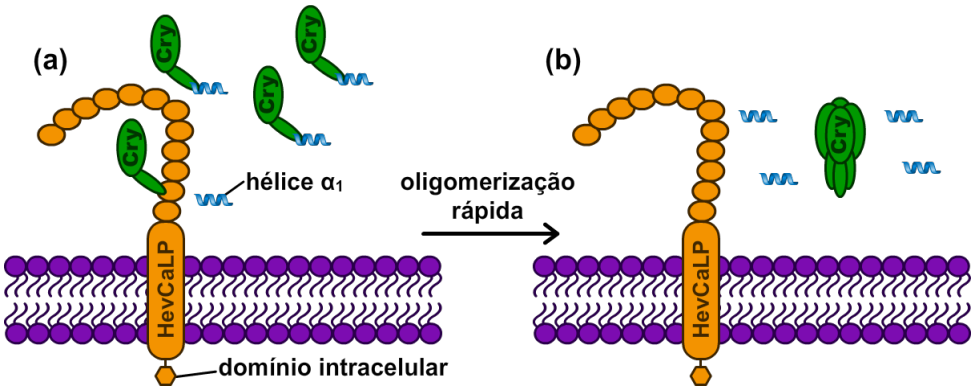
- Lebel et al. (2009) estabeleceram que o Domínio I é essencial na formação dos poros, visto que mutações nesse domínio pioram drasticamente a taxa de formação dos mesmos;
- Leetachewa et al. (2014) determinaram que os resíduos ^{157}R^{159} e ^{169}Y^{171} são importantes para a manifestação tóxica da proteína Cry4Ba;
- Zhang et al. (2014) afirmam que mutações na CADR conferem resistência à proteína Cry1Ac;
- Segundo Elleuch et al. (2015), a proteína Cyt1Aa também tem ação de sinergismo com a proteína Cry4Ba no chamado mosquito da dengue (*Aedes aegypti*) aumentando a sua potência;
- De acordo com Feng et al. (2015), o Dominio III da proteína Cry1Ie é importante na sua manifestação tóxica na traça do milho asiática (*Ostrinia furnacalis*);
- Qiu et al. (2015) demonstram que a CADR também é um receptor importante para a proteína Cry2Aa;
- Shu et al. (2015) afirmam que a toxicidade da proteína Cry8Ea não envolve os receptores CADR, APN e/ou ALP;
- Segundo Zhang, Hua e Adang (2015), a caderina também é um dos receptores da proteína Cry11Ba;
- Zuniga-Navarrete et al. (2015) afirmam que o *loop* 1 do Domínio II da proteína Cry3Aa tem como sítio de ligação uma região na repetição 12 da caderina de *Tenebrio molitor*, conhecido como bicho-da-farinha;
- Jin et al. (2016) identificaram que a ALP é um possível receptor da proteína Cry1Ac na traça do milho asiática (*Ostrinia furnacalis*);
- Endo et al. (2017) demonstraram que os transportadores ABC dos tipos C2 e C3 (ABCC2 e ABCC3) da ordem *Lepidoptera* são ativos com a proteína Cry1Aa, mas não às proteínas Cry1Ca e Cry1Da. Afirmam também que o transportador ABCC da ordem *Coleoptera* apresenta especificidade à proteína Cry8Ca, além de que o mesmo transportador da ordem *Diptera* não apresenta atividade com as proteínas Cry ativas contra lepidópteros e dípteros.

No trabalho de Bretschneider, Heckel e Pauchet (2016) é apresentado um esquema gráfico, adaptado e reproduzido neste trabalho nas Figuras 54, 55, 56 e 57, do possível modo de ação dos receptores HevCaLP (caderina) e HevABCC2 (transportador ABC) que foi proposto por Zhang et al. (2012). Os componentes que formam essas figuras estão

coloridos da seguinte forma: os monômeros e os oligômeros da proteína Cry1A em verde, a hélice α_1 em azul, a membrana celular em roxo e os receptores HevCaLP e HevABCC2 respectivamente em amarelo e vermelho.

Na Figura 54 é apresentado o funcionamento do receptor HevCaLP no processo de oligomerização rápida das proteínas Cry1A. Nesse processo, os monômeros de Cry1A associam-se ao receptor que, por sua vez, atua na clivagem da hélice α_1 dos monômeros (Figura 54a). A partir de várias toxinas clivadas, são formados na solução os oligômeros pré-poro das proteínas Cry1A, que constituem a estrutura responsável pela posterior formação do poro (Figura 54b).

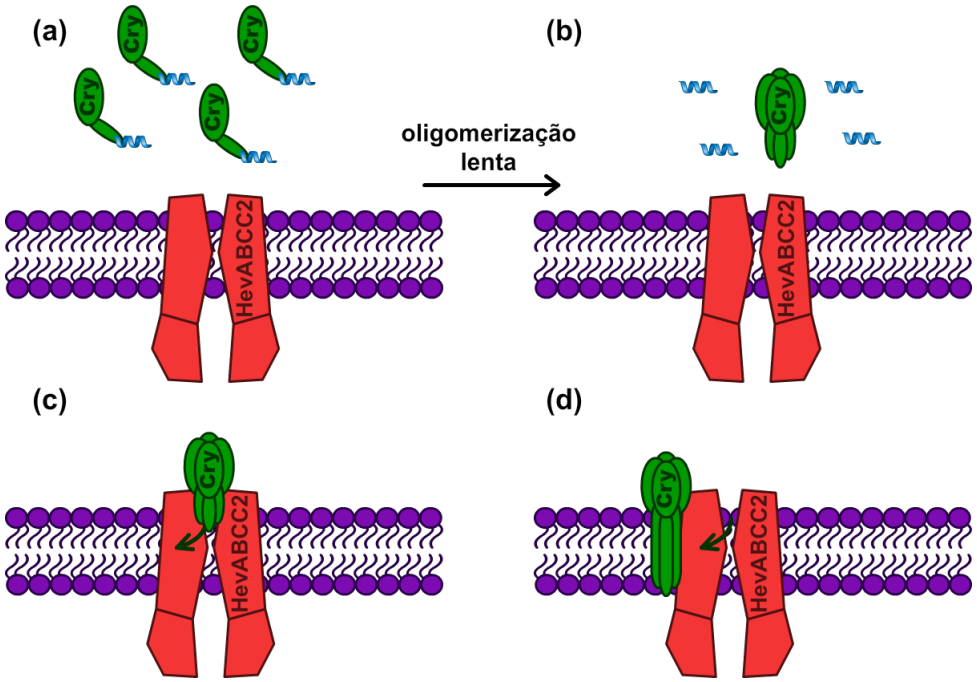
Figura 54 – Modo de ação da proteína Cry1A proposto por Zhang et al. (2012) - HevCaLP



Fonte: Adaptado de Bretschneider, Heckel e Pauchet (2016) pelo autor

A oligomerização lenta, ou seja, a criação de oligômeros da proteína Cry1A sem a atuação catalizadora de HevCaLP, também pode ocorrer. Esse cenário é apresentado na Figura 55, em que é mostrado na Figura 55a as toxinas com a hélice α_1 ainda não clivada e na Figura 55b a formação de um oligômero. Na Figura 55c é apresentada a ligação do oligômero pré-poro com o receptor HevABCC2 e, por fim, na Figura 55d o oligômero é apresentado de forma ligada ao receptor HevABCC2, sem ter sido inserido na membrana.

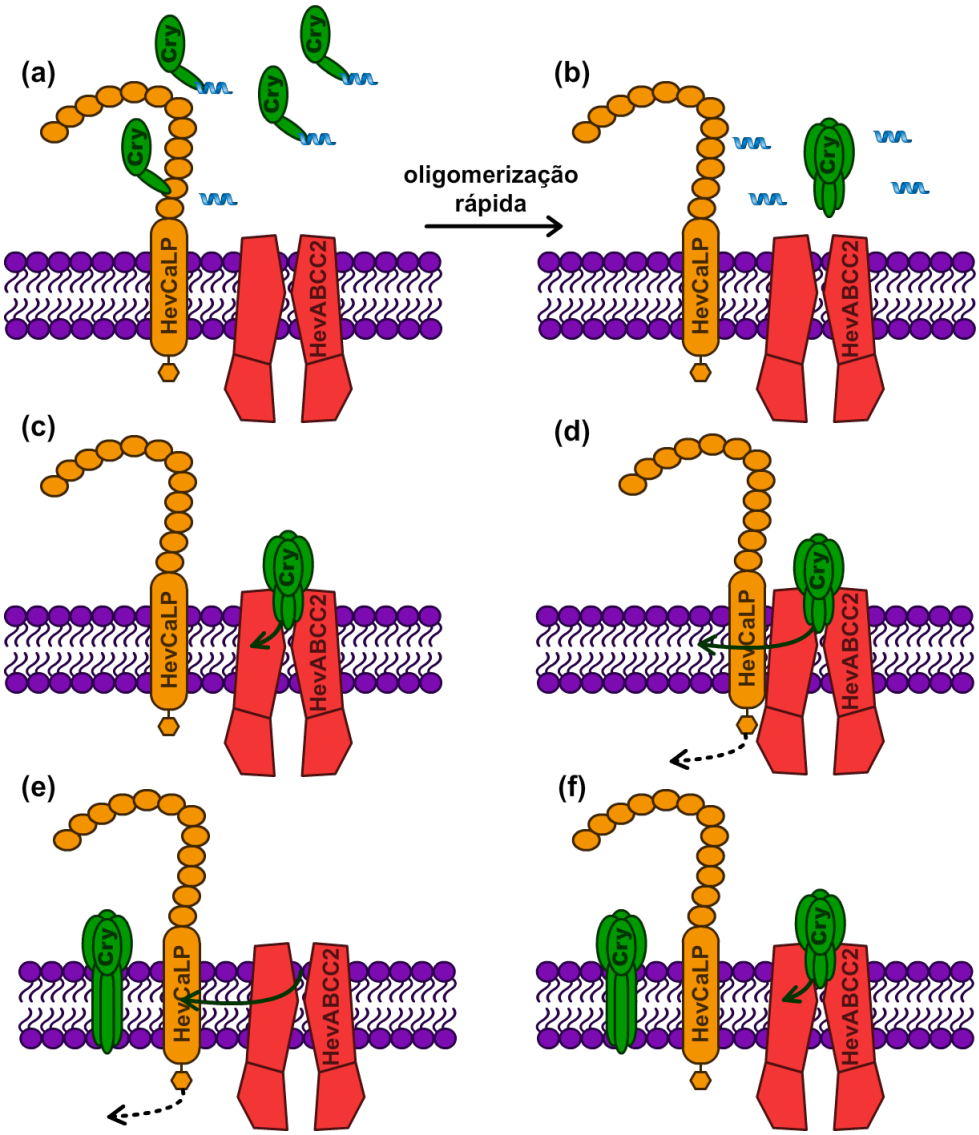
Figura 55 – Modo de ação da proteína Cry1A proposto por Zhang et al. (2012) - HevABCC2



Fonte: Adaptado de Bretschneider, Heckel e Pauchet (2016) pelo autor

Na Figura 56 é apresentado o cenário em que há a interação com os dois receptores. Na Figura 56a, os monômeros de Cry1A interagem com o receptor HevCaLP que, por sua vez, atuam na clivagem da hélice α_1. Na Figura 56b é apresentado o processo de formação dos oligômeros pré-poro. Na Figura 56c o oligômero pré-poro se liga ao receptor HevABCC2 e na Figura 56d o oligômero que está ligado ao receptor HevABCC2 se liga ao receptor HevCaLP (seta em verde). Na Figura 56e o oligômero é removido do receptor HevABCC2 por meio de interações com o citoesqueleto, representado pela seta tracejada colorida em preto. Por fim, na Figura 56f, é apresentada a continuação do processo de inserção dos oligômeros na membrana, em que o próximo oligômero interage com o receptor HevABCC2.

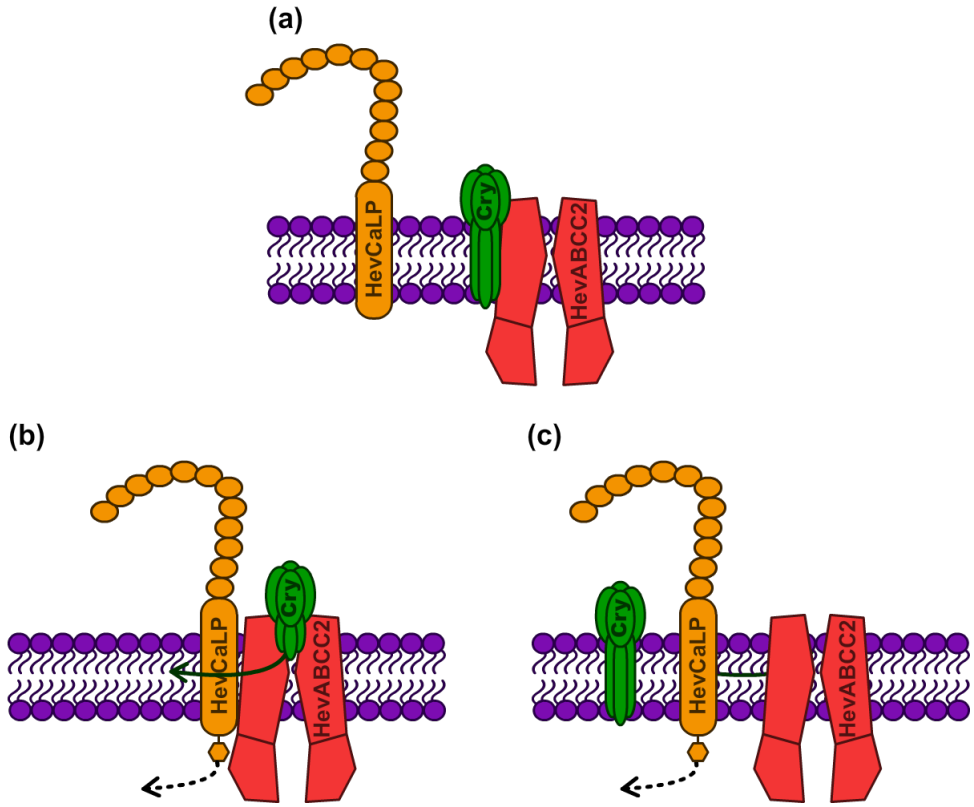
Figura 56 – Modo de ação da proteína Cry1A proposto por Zhang et al. (2012) - HevCaLP + HevABCC2



Fonte: Adaptado de Bretschneider, Heckel e Pauchet (2016) pelo autor

Por fim, na Figura 57 é apresentada a importância do domínio intracelular de HevCaLP na fixação do oligômero pré-poro na membrana celular. Na Figura 57a é mostrado o cenário em que o receptor HevCaLP não possui o domínio intracelular, fazendo com que o oligômero não possa ser removido do receptor HevABCC2 e inserido na membrana. Na Figura 57b e Figura 57c é apresentada a remoção do oligômero e inserção na membrana por meio de interações com o citoesqueleto, mediada pelo domínio intracelular de HevCaLP.

Figura 57 – Modo de ação da proteína Cry1A proposto por Zhang et al. (2012) - Envolvimento do domínio intracelular de HevCaLP



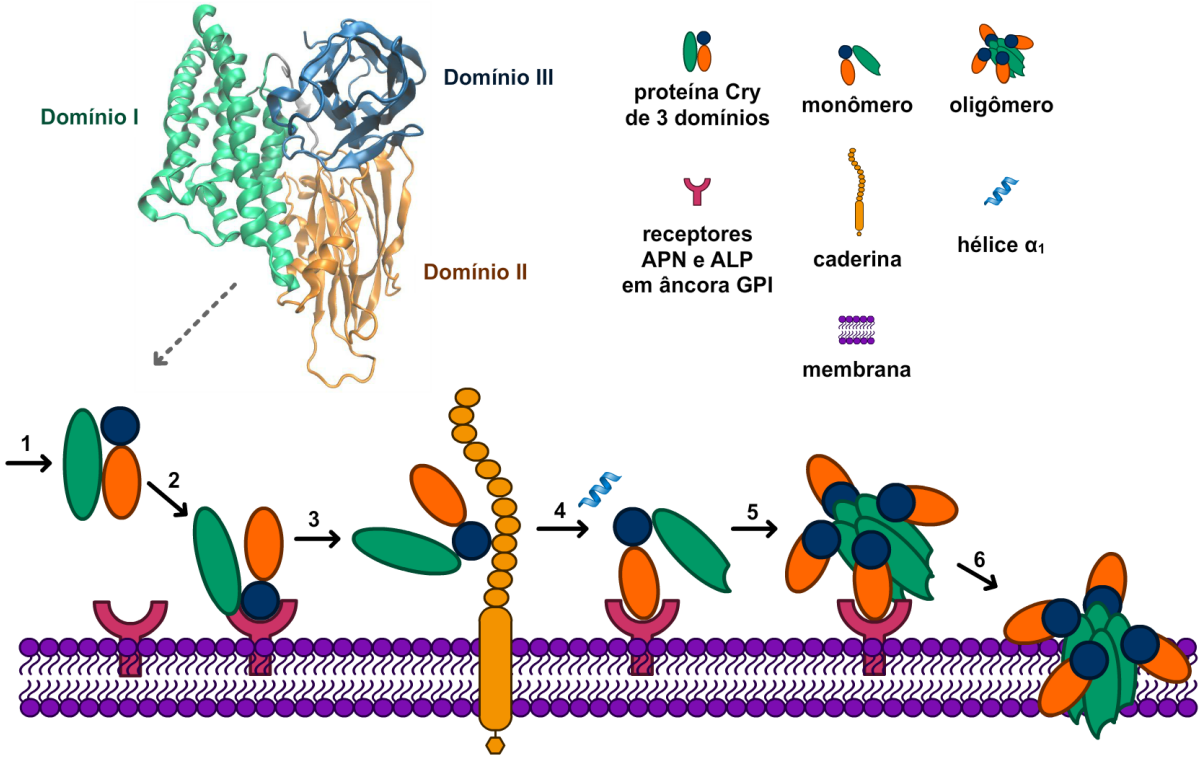
Fonte: Adaptado de Bretschneider, Heckel e Pauchet (2016) pelo autor

Xu et al. (2014) apresentam outro esquema para representar o modo de ação da proteínas Cry1A. Na Figura 58 esse esquema é apresentado. Acima e à esquerda é apresentada a estrutura da proteína Cry1Aa1 (PDB: 1CIY), com o Domínio I colorido em verde, o Domínio II colorido em laranja e o Domínio III colorido em azul. Acima e à direita é apresentada a legenda dos componentes que formam o diagrama apresentado abaixo. Nesse diagrama são apresentados seis passos que representam a formação do poro na membrana celular:

1. A proteína Cry1Aa é solubilizada e digerida no intestino do inseto;
2. A proteína liga-se aos receptores APN e ALP contidos na membrana celular;
3. A ligação da proteína na caderina facilita a clivagem proteolítica da hélice α_1 da extremidade n-terminal (Domínio I);
4. A remoção da hélice α_1 resulta no monômero, o qual tem sua afinidade de ligação com os receptores APN e ALP aumentada, induzindo a formação do oligômero pré-poro;
5. Formação do oligômero pré-poro;

6. Inserção irreversível do oligômero pré-poro na membrana, induzindo a lise celular.

Figura 58 – Modo de ação da proteína Cry1A apresentado por Xu et al. (2014)



Fonte: Adaptado de Xu et al. (2014) pelo autor

Pelo exposto, pode-se notar que existem diversos trabalhos que têm procurado elucidar a questão de quais são os receptores envolvidos na toxicidade das proteínas Cry, bem como qual é o funcionamento dos mesmos, mas a grande maioria desses trabalhos têm investigado o modo de ação das proteínas da família Cry1A em insetos da ordem *Lepidoptera*. Outro ponto importante na pesquisa das proteínas Cry é relacionado à obtenção dos modelos tridimensionais das mesmas. Sendo assim, na próxima Seção serão apresentados os modelos das proteínas Cry que estão disponíveis de forma pública atualmente.

2.3.4 Modelos

Existem diversos modelos de proteínas Cry descritos na literatura e depositados no PDB e no *Protein Model Database* (PMDB)[23], entretanto há modelos que foram depositados e ainda não têm um trabalho publicado, bem como trabalhos publicados que descrevem modelos que foram supostamente gerados, mas que não estão acessíveis nos bancos de dados estruturais citados e nem mesmo foram disponibilizados pelos seus autores em outro meio eletrônico (BUZATTO; FRANÇA; ZINGARETTI, 2016). Na Tabela 5 são apresentados todos estes detalhes, informando as proteínas, as ordens que são afetadas, seus modelos, os trabalhos publicados e algumas observações, quando necessário.

[23] O PMDB foi desenvolvido por Castrignano et al. (2006).

Tabela 5 – Modelos das proteínas Cry

Proteína	**Ordem(s) Afetadas**	**Identificador do Modelo**	**Referência(s)**	**Observações**
Cry1Aa1	*Diptera*, *Lepidoptera* e *Gastropoda*	1CIY	Knowles e Ellar (1987) e Grochulski et al. (1995)	
Cry1Ab16	*Diptera*, *Lepidoptera* e *Gastropoda*	Indisponível	Kashyap (2012)	Não há modelo depositado
Cry1Ab19	*Diptera*, *Lepidoptera* e *Gastropoda*	Indisponível	Kashyap, Singh e Amla (2012)	Não há modelo depositado
Cry1Ac1	*Diptera*, *Lepidoptera* e *Gastropoda*	4ARX 4ARY 4W8J	Ainda não publicado Ainda não publicado Derbyshire, Ellar e Li (2001)	
Cry1Ld	*Lepidoptera*	Indisponível	Dehury et al. (2013)	Não há modelo depositado
Cry2Aa1	*Diptera*, *Hemiptera* e *Lepidoptera*	1I5P	Morse, Yamamoto e Stroud (2001)	
Cry3A	*Coleoptera*, *Hemiptera* e *Hymenoptera*	1DLC	Li, Carroll e Ellar (1991)	
Cry3Aa1	*Coleoptera*, *Hemiptera* e *Hymenoptera*	4QX0 4QX1 4QX2 4QX3	Sawaya et al. (2014)	
Cry3Bb1	*Coleoptera*	1JI6	Galitsky et al. (2001)	
Cry4Aa1	*Diptera*	2C9K	Boonserm, Angsuthanasombat e Lescar (2004) e Boonserm et al. (2006)	
Cry4Ba1	*Diptera*	1W99 4MOA	Boonserm et al. (2005) Sriwimol et al. (2015)	
Cry5Aa1	*Hymenoptera* e *Rhabditida*	PM0074964	Xin-Min et al. (2009)	
Cry5B	*Rhabditida*	4D8M	Hui et al. (2012)	
Cry5Ba1	*Rhabditida*	PM0075036	Xia et al. (2008)	
Cry6Aa	*Rhabditida*	5J66 5KUC 5KUD	Dementiev et al. (2016)	Formação de poro
Cry6Aa2	*Rhabditida*	5GHE	Huang et al. (2016)	Formação de poro
Cry8Ea1	*Coleoptera*	3EB7	Guo et al. (2009)	
Cry11Bb1	*Diptera*	Indisponível	Gutierrez, Alzate e Orduz (2001)	Não há modelo depositado
Cry23Aa1 Cry37Aa1	*Coleoptera*	4RHZ	Ainda não publicado	Complexo proteico binário
Cry30Ca2	*Diptera*	Indisponível	Zhao, Zhou e Xia (2012)	Não há modelo depositado
Cry34Ab1	*Coleoptera*	4JOX	Kelker et al. (2014)	
Cry35Ab1	*Coleoptera*	4JP0	Kelker et al. (2014)	
Cry51Aa1	*Coleoptera* e *Hemiptera*	4PKM	Xu et al. (2015)	
Cry51Aa2	*Coleoptera* e *Hemiptera*	5HD2	Gowda et al. (2016)	

* Os identificadores de modelo que possuem códigos com quatro caracteres representam identificadores do PDB, enquanto os identificadores com 9 caracteres são códigos do PMDB.

Fonte: Adaptado de Buzatto, França e Zingaretti (2016)

Pode-se notar que foram obtidos diversos modelos das proteínas Cry, visto que as mesmas têm importância comercial relevante e, por isso, esse viés comercial será apresentado na próxima Seção.

2.3.5 Plantas Transgênicas e Produtos de *Bt*

O uso das proteínas Cry como inseticida correspondia, até o ano de 2009, a 98% dos biopesticidas utilizados, tanto na agricultura quanto na saúde (FIÚZA; BERLITZ, 2009), sendo aplicados diretamente nas culturas suscetíveis aos insetos-praga. Em 1981, Schnepf e Whiteley foram responsáveis em caracterizar e clonar os genes da proteína Cry, permitindo-se vislumbrar novas aplicações do *Bt*. Uma dessas aplicações é a transgenia, que permitiu inserir os genes codificadores das proteínas Cry em genomas vegetais, fazendo com as plantas pudessem produzir, além de suas toxinas normais (quando existiam), as proteínas Cry, aumentando assim sua resistência aos insetos-praga. Na Tabela 6 estão apresentados alguns biopesticidas comerciais que utilizam o *Bt*, enquanto na Tabela 7 encontram-se listadas algumas plantas modificadas geneticamente para a produção de proteínas Cry. Mais exemplos de produtos e plantas transgênicas podem ser encontrados nos trabalhos de Fiúza e Berlitz (2009), Fiúza e Pinto (2009) e Sanahuja et al. (2011). Além disso, no trabalho de Raymond e Federici (2017) pode ser encontrado um estudo sobre a segurança da utilização de produtos de *Bt*.

Tabela 6 – Alguns biopesticidas comerciais de *Bt* para controle de pragas agrícolas

Produto	**Empresa**	**Cepa do *Bt***	**Ordem Afetada**
Dipel	Abbott	*Bt kurstaki*	*Lepidoptera*
Thuricide	Sandoz	*Bt kurstaki*	*Lepidoptera*
XenTari	Abbott	*Bt aizawai*	*Lepidoptera*
M-One	Mycogen	*Bt san diego* e *Bt tenebrionis*	*Coleoptera*
Di-Terra	Abbott	*Bt san diego* e *Bt tenebrionis*	*Coleoptera*

Fonte: Adaptado de Fiúza e Berlitz (2009) pelo autor

Tabela 7 – Variações transgênicas de algumas plantas que sintetizam proteínas Cry ativas contra a ordem *Lepidoptera*

Planta	**Gene da Proteína**	**Inseto(s)-Alvo**
Amendoim	Cry1Ac	*Elasmopalpus lignosellus*
Arroz	Cry1Ab	*Chilo suppressalis* *Cnaphalocrocis medinalis* *Herpetogramma licarisalis* *Mycalesis gotama* *Naranga canescens* *Parnara guttata* *Scirpophaga incertulas* *Sesamia inferens*
Batata	Cry1Ab	*Phthorimaea operculella*
Milho	Cry1F	*Diatraea grandiosella* *Ostrinia nubilalis*

Fonte: Adaptado de Fiúza e Pinto (2009) pelo autor

Pelo exposto nesse Capítulo, nota-se a importância das proteínas como sendo os blocos estruturais básicos para a "construção" de todos os organismos vivos e que as proteínas Cry, sintetizadas pelo *Bt*, têm importância comercial relevante, visto seu potencial de aplicação na agroindústria como um biopesticida altamente específico (FENG et al., 2015; KOCH et al., 2015). Nos próximos Capítulos serão apresentadas a Hipótese e os Objetivos desta pesquisa.

3 HIPÓTESE

A hipótese desta tese é que as modificações na estrutura terciária das proteínas Cry estão relacionadas à especificidade destas no controle de diferentes ordens de insetos.

4 OBJETIVOS

4.1 OBJETIVO GERAL

Identificar quais as diferenças conformacionais das proteínas Cry que estão associadas à sensibilidade tóxica das ordens de insetos suscetíveis.

4.2 OBJETIVOS ESPECÍFICOS

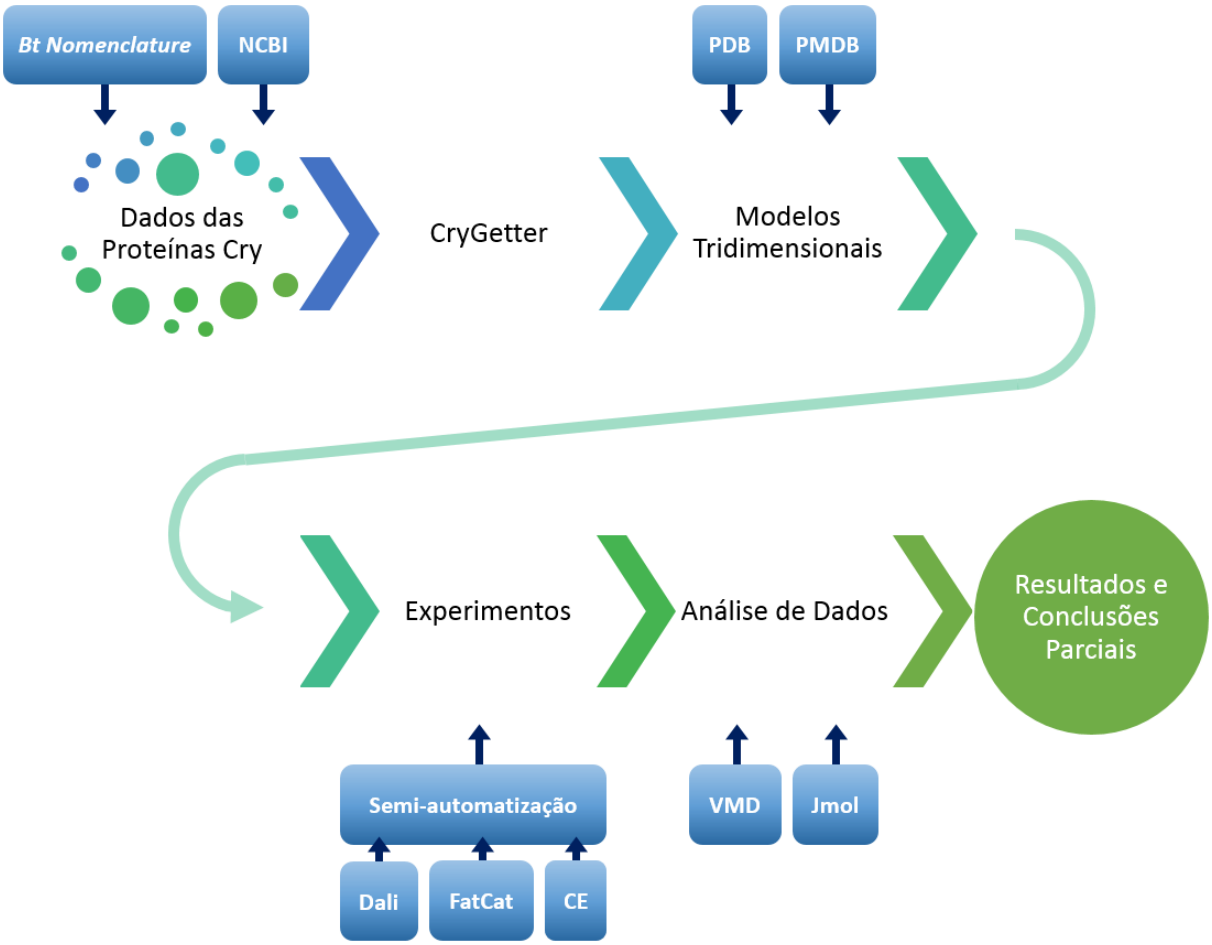
Os objetivos específicos são:

- Obter os dados das estruturas primárias das proteínas Cry conhecidas;
- Seccionar as estruturas primárias das proteínas Cry obtidas nos três domínios que as constituem;
- Utilizar os dados obtidos (sequências e domínios) na obtenção de modelos tridimensionais que representam as estruturas das proteínas;
- Utilizar algoritmos e/ou ferramentas para o alinhamento estrutural (tridimensional) dos modelos obtidos;
- Renderizar os alinhamentos estruturais obtidos em imagens para inspeção visual;
- Buscar e analisar padrões nas imagens geradas a partir dos modelos;
- Analisar os dados obtidos, visando identificar os padrões que diferenciam cada proteína com relação às ordens de insetos atacadas, focando especificamente em regiões ativas com os receptores dos organismos alvo.

5 METODOLOGIA

Neste trabalho foram desenvolvidas diversas atividades com intuito de alcançar os objetivos traçados, sendo que o caminho percorrido para o desenvolvimento dessas atividades está organizado de forma cronológica no diagrama apresentado na Figura 59.

Figura 59 – Caminho percorrido



Fonte: Elaborada pelo autor

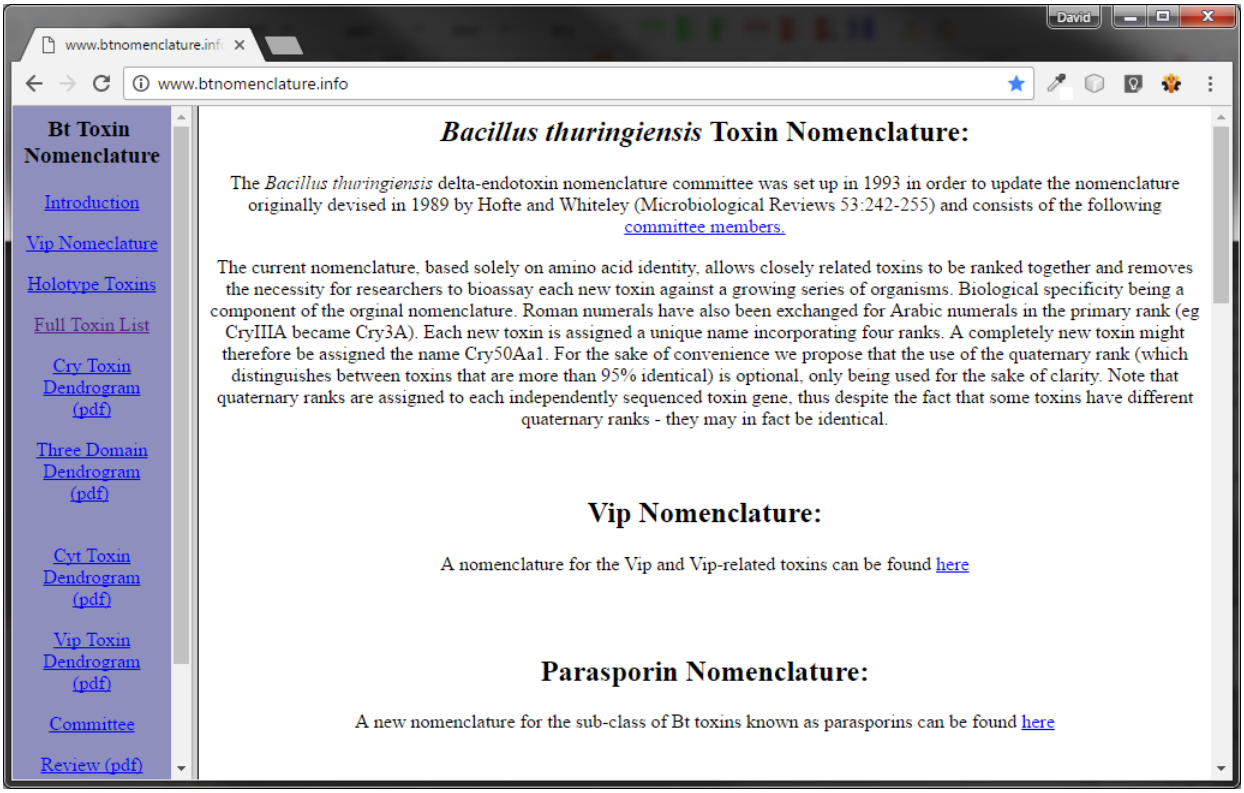
Esse diagrama deve ser lido da esquerda para a direita, de cima para baixo, sendo que a primeira atividade está identificada como “Dados das Proteínas Cry” e a última como “Resultados e Conclusões Parciais”. Além do fluxo principal das atividades, pode-se notar algumas caixas, representadas em azul, que identificam recursos e/ou tecnologias que foram utilizados no apoio à execução de uma atividade específica. Por exemplo, na primeira atividade, “Dados das Proteínas Cry”, são usados recursos do site *Bt Nomenclature* (CRICKMORE et al., 2016), criado e mantido por Crickmore et al., em que as proteínas Cry que foram publicadas estão catalogadas, e do NCBI, por meio de uma *Application Programming Interface* (API). Nas próximas Seções, cada uma dessas atividades será

detalhada.

5.1 DADOS DAS PROTEÍNAS CRY

Os dados das proteínas Cry descobertas e sequenciadas podem ser obtidos diretamente no NCBI, entretanto, Crickmore et al. (2016), têm mantido o site *Bt Nomenclature*, que atua como um catálogo das proteínas Cry. Neste trabalho, esse site foi usado como ponto de partida para o entendimento sobre essas proteínas. Na Figura 60 é apresentada uma captura de tela da página principal do site *Bt Nomenclature*, em que é apresentado um menu ao lado esquerdo, com as seções do site, e do lado direito a apresentação do mesmo.

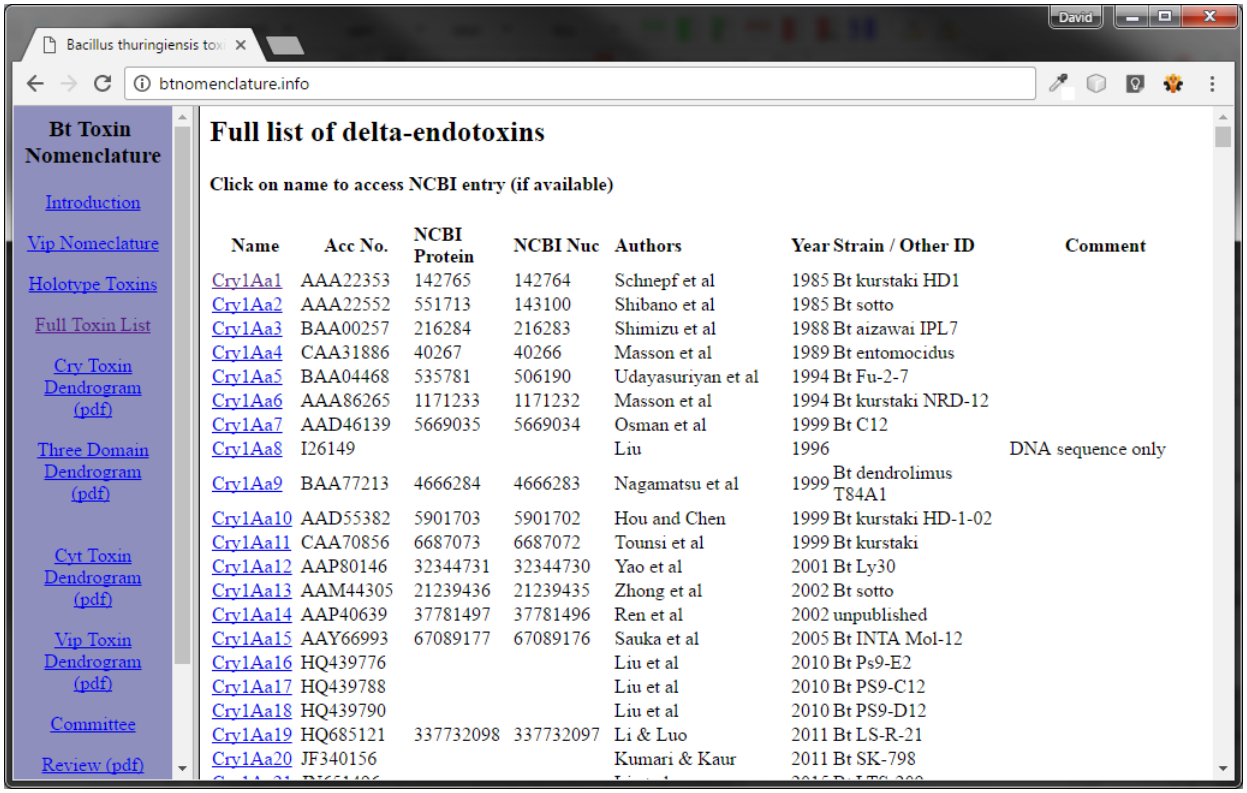
Figura 60 – Página principal do site *Bt Nomenclature*



Fonte: Captura de tela da página principal do site *Bt Nomenclature* (CRICKMORE et al., 2016)

O catálogo das toxinas, que é mantido pelos autores, é acessado ao se clicar no item "*Full Toxin List*" do menu à esquerda. Essa lista é armazenada no site por meio de uma tabela *Hypertext Markup Language* (HTML), que pode ser vista na Figura 61.

Figura 61 – Lista de toxinas catalogadas do site *Bt Nomenclature*

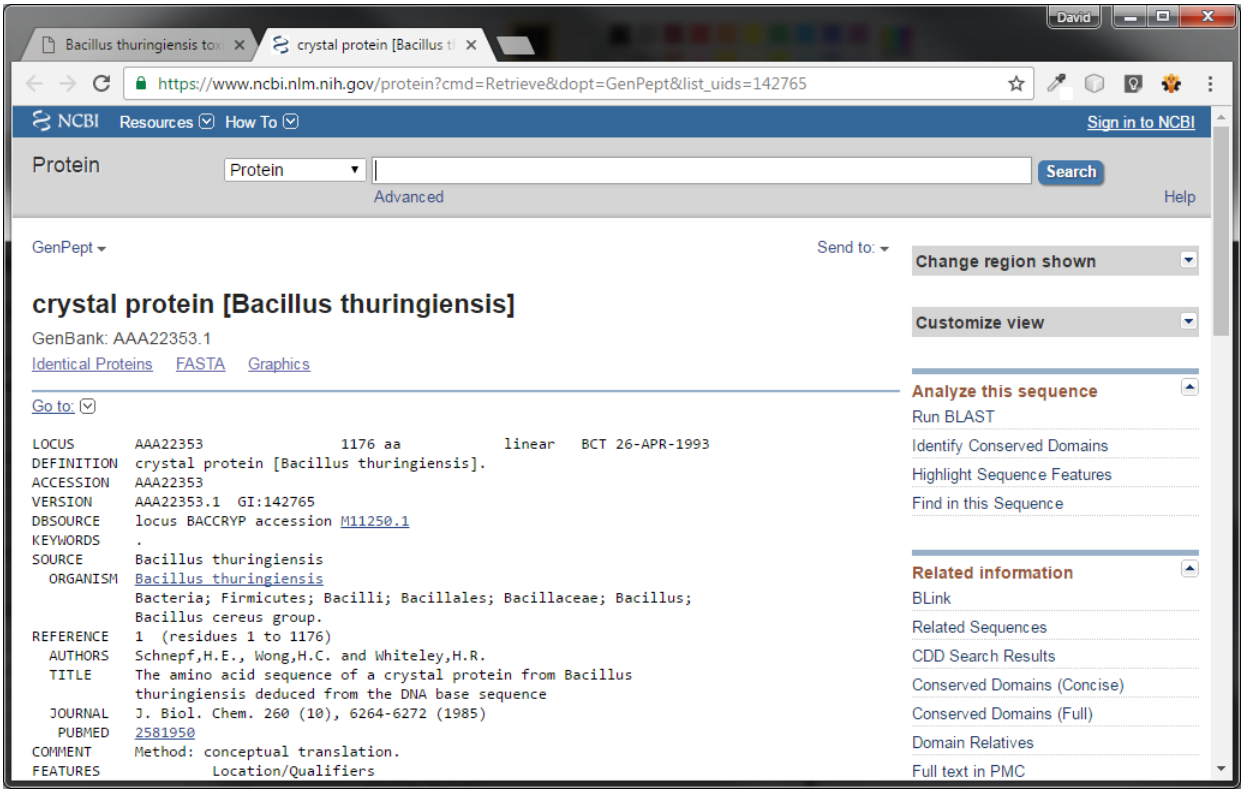


Fonte: Captura de tela da página de listagem de toxinas catalogadas do site *Bt Nomenclature* (CRICKMORE et al., 2016)

Essa tabela está organizada de modo a apresentar os principais identificadores de cada proteína catalogada. Por exemplo, na primeira linha, na coluna "*Name*", é apresentado o nome da proteína, de acordo com a classificação de Crickmore et al. (1998) e, que nesse caso, é Cry1Aa1. Nessa mesma coluna, o nome da proteína é um *link* que, ao ser clicado, redireciona o usuário do site para a página de dados da proteína no NCBI. Na segunda coluna, "*Acc. No.*", é apresentado o *accession number*, um identificador único que sinaliza em qual banco de dados a entrada foi depositada. Na terceira coluna, "NCBI *Protein*", é apresentado o identificador da entrada no banco de dados de proteínas do NCBI e, de forma análoga, na quarta coluna, "NCBI *Nucleotide*", é apresentado o identificador do banco de dados de nucleotídeos do NCBI. Na quinta coluna, "*Authors*", são apresentados os autores do trabalho que originou o depósito dos dados, na sexta coluna, "*Year Strain / Other ID*", são apresentados os dados relativos à cepa que sintetiza a proteína, ou seja, de qual cepa do *Bt* a proteína foi isolada e, por fim, na sétima coluna, "*Comment*", são apresentados alguns comentários que os mantenedores do site julgam ser pertinentes.

Ao clicar no item discutido anteriormente, o usuário pode visualizar os dados da entrada no NCBI. Na Figura 62 é apresentada uma captura de tela da interface do NCBI em que são mostrados os detalhes da proteína Cry1Aa1.

Figura 62 – Dados da proteína Cry1Aa1 no NCBI



Fonte: Captura de tela da página de dados da proteína Cry1Aa1 no NCBI (CRYSTAL. . . , 2016)

Para que fosse feita a coleta de todos os dados das proteínas catalogadas, haveria a necessidade de se visitar manualmente cada um dos *links* de uma tabela que contém centenas de linhas. Percebeu-se que esse processo manual poderia contaminar os dados que seriam obtidos, pois ao se executar essa tarefa manualmente, a chance de se copiar dados truncados, ou seja, selecionar as sequências de aminoácidos de forma incompleta, ou mesmo armazenar dados de forma errada, por exemplo, salvar os dados da proteína Cry2Aa1 em um arquivo que deveria conter os dados da proteína Cry1Aa1, seria enorme.

Dada à existência do site *Bt Nomenclature* e da possibilidade de se obter a tabela de toxinas catalogadas de forma remota, ou seja, por meio de uma aplicação computacional que executaria essa tarefa remotamente, vislumbrou-se a possibilidade de automatizar o processo de coleta de dados de cada proteína, tornando essa atividade menos propensa a erros. Com isso, foi dado início ao desenvolvimento de uma ferramenta computacional que teria como objetivo executar essa coleta, além de permitir que os usuários da mesma fossem capazes de visualizar os dados de cada proteína em um ambiente único, fornecendo também recursos para processar as sequências das proteínas com o uso de algoritmos de alinhamento de sequências, entre outras funcionalidades. Os detalhes da arquitetura e do funcionamento dessa ferramenta serão apresentados na próxima Seção.

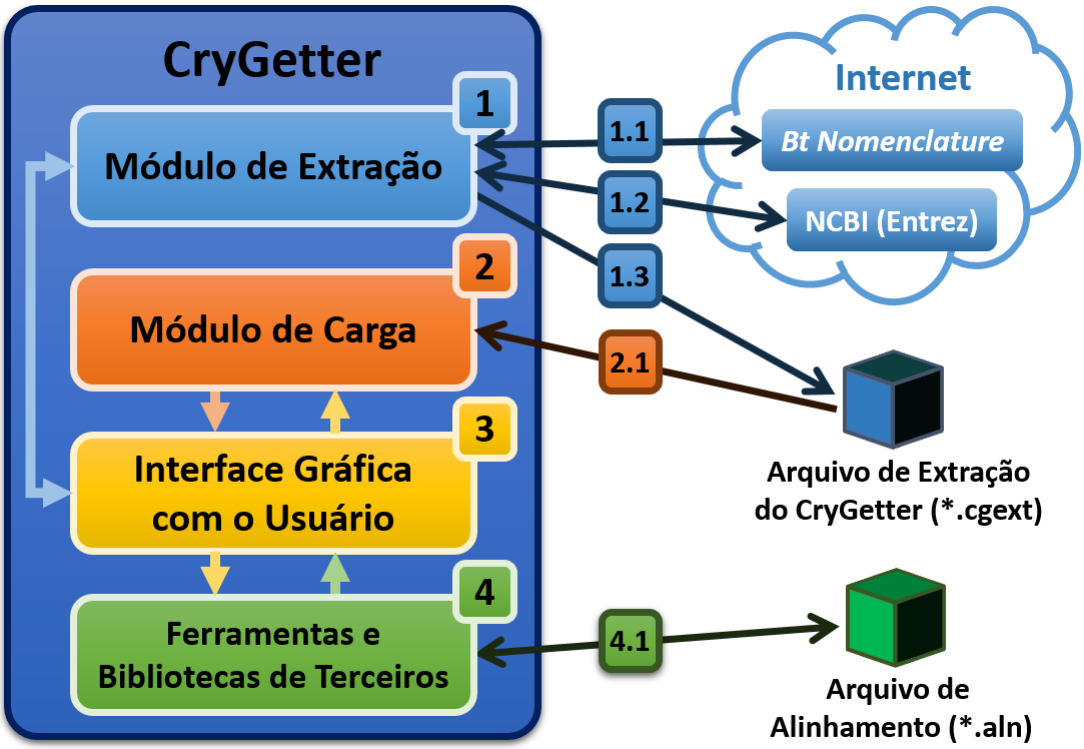
5.2 CRYGETTER

A ferramenta CryGetter foi criada inicialmente como meio de auxiliar no aprendizado sobre as proteínas Cry. Essa ferramenta é capaz de compilar os dados das proteínas Cry catalogadas no site *Bt Nomenclature* e integrá-los com dados obtidos no NCBI. Seu desenvolvimento é relatado no trabalho de Buzatto, França e Zingaretti (2016), sendo que os detalhes desse processo são apresentados a seguir.

Do ponto de vista de implementação, ou seja, da construção da ferramenta utilizando uma linguagem de programação de computadores, o CryGetter foi desenvolvido utilizando a linguagem Java, mais especificamente a versão 8 da plataforma, além de fazer uso de diversas ferramentas e bibliotecas de terceiros que têm o objetivo de realizar alguns processamentos específicos, como o alinhamento das sequências das proteínas, a visualização dos alinhamentos computados, a renderização tridimensional de moléculas, entre outros.

Como entrada, a ferramenta utiliza os dados da lista de toxinas do site *Bt Nomenclature*, bem como as ligações externas que essa fonte de dados tem com o NCBI, que por sua vez são processadas de modo a obter os dados de cada proteína catalogada utilizando o serviço de consultas *Global Query Cross-Database Search System* (Entrez) (MLA. . . , 2016), também do NCBI. A arquitetura do CryGetter pode ser vista na Figura 63.

Figura 63 – Arquitetura do CryGetter

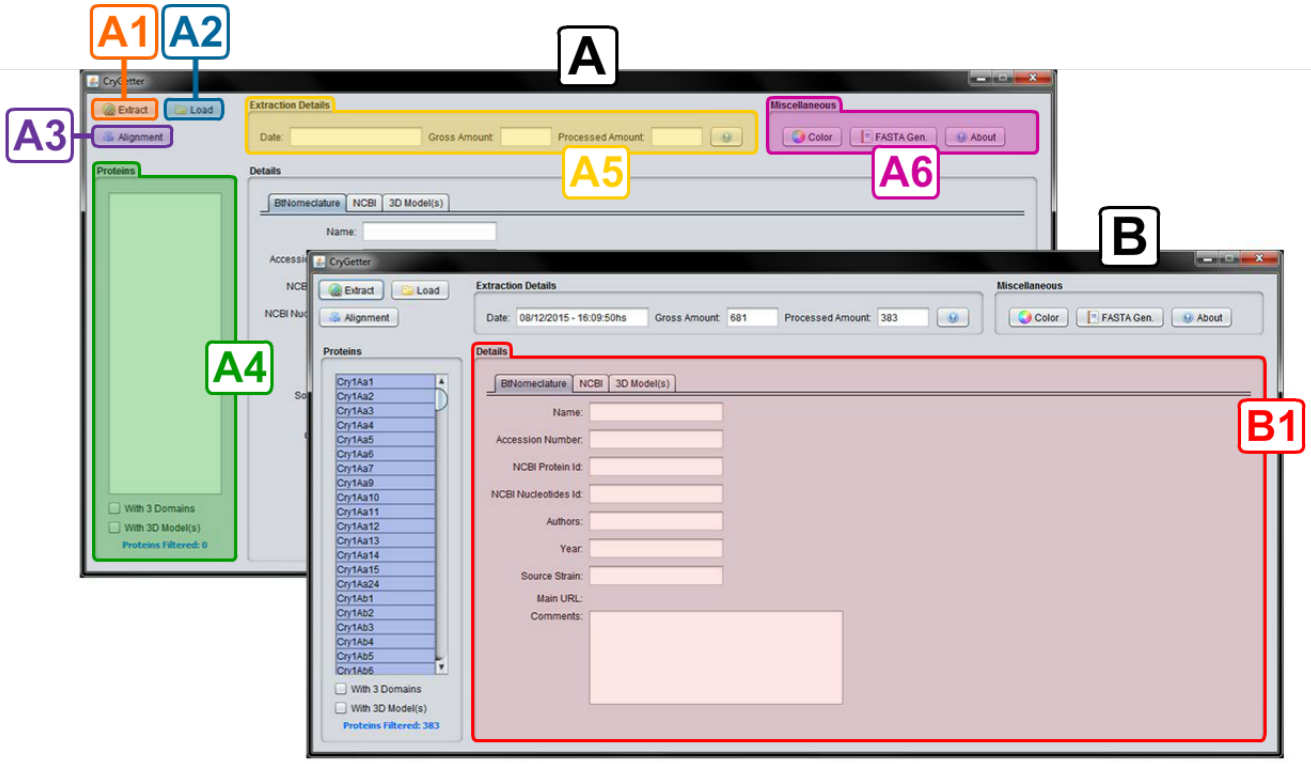


Fonte: Adaptada de Buzatto, França e Zingaretti (2016)

Na Figura 63, representado em azul claro e pelo número 1, é apresentado o "Módulo de Extração" do CryGetter, que é executado pela ferramenta ao se clicar no botão "*Extract*".

Esse botão está destacado na seção A1 (em laranja) da Figura 64 em que, por sua vez, está representada a interface gráfica principal da ferramenta.

Figura 64 – Interface gráfica principal do GryGetter



Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

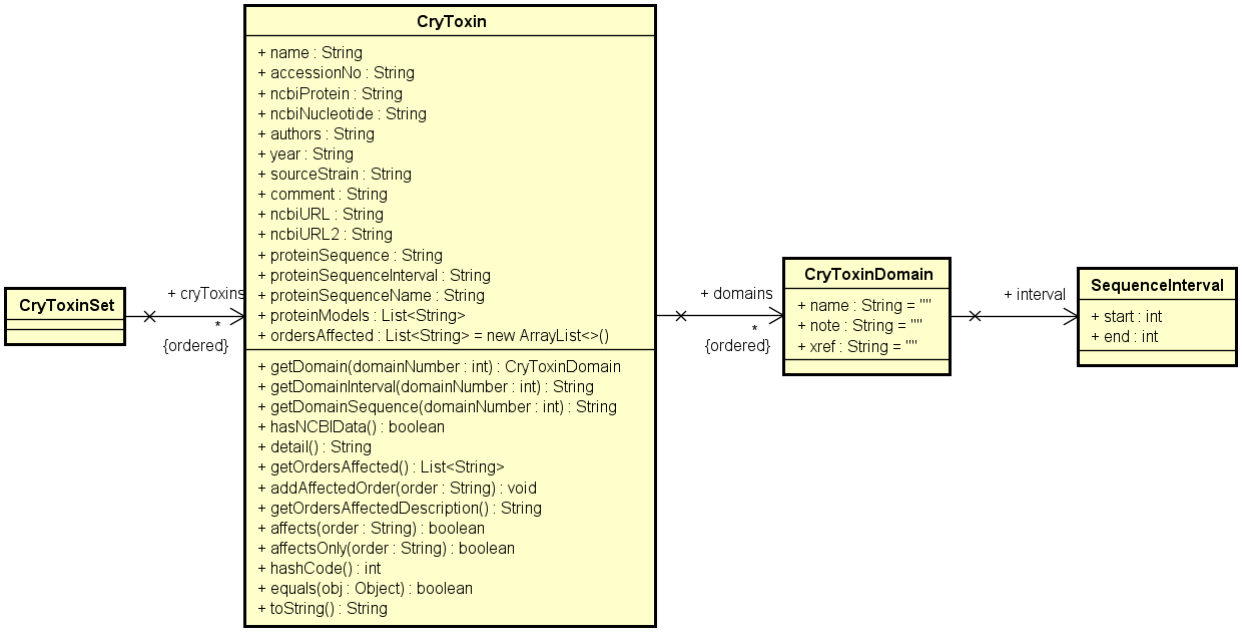
O funcionamento do módulo de extração é dividido em quatro passos:

1. **Extração de dados do site *Bt Nomenclature*:** O módulo de extração requisita e obtém os dados de um arquivo HTML específico[1] do site *Bt Nomenclature*, sendo que esse processo é apresentado pela seta 1.1 da Figura 63. É nesse arquivo que a tabela que contém o catálogo das proteínas Cry está inserida;

2. **Pré-processamento dos dados das proteínas Cry:** A partir dos dados não processados obtidos no passo anterior, o módulo de extração os analisa e cria uma lista encadeada de um Tipo Abstrato de Dados (TAD) chamado "CryToxin", que por sua vez contém os dados de cada linha da tabela do catálogo. Na Figura 65 é apresentado o diagrama de classes da *Unified Modeling Language* (UML) que representa a composição usada para apoiar no processo de serialização em *Extensible Markup Language* (XML) e deserialização em objetos Java dos dados contidos no documento HTML;

[1] Disponível no endereço http://www.lifesci.sussex.ac.uk/home/Neil_Crickmore/Bt/toxins2.html

Figura 65 – Diagrama de classes do TAD “CryToxin”



Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

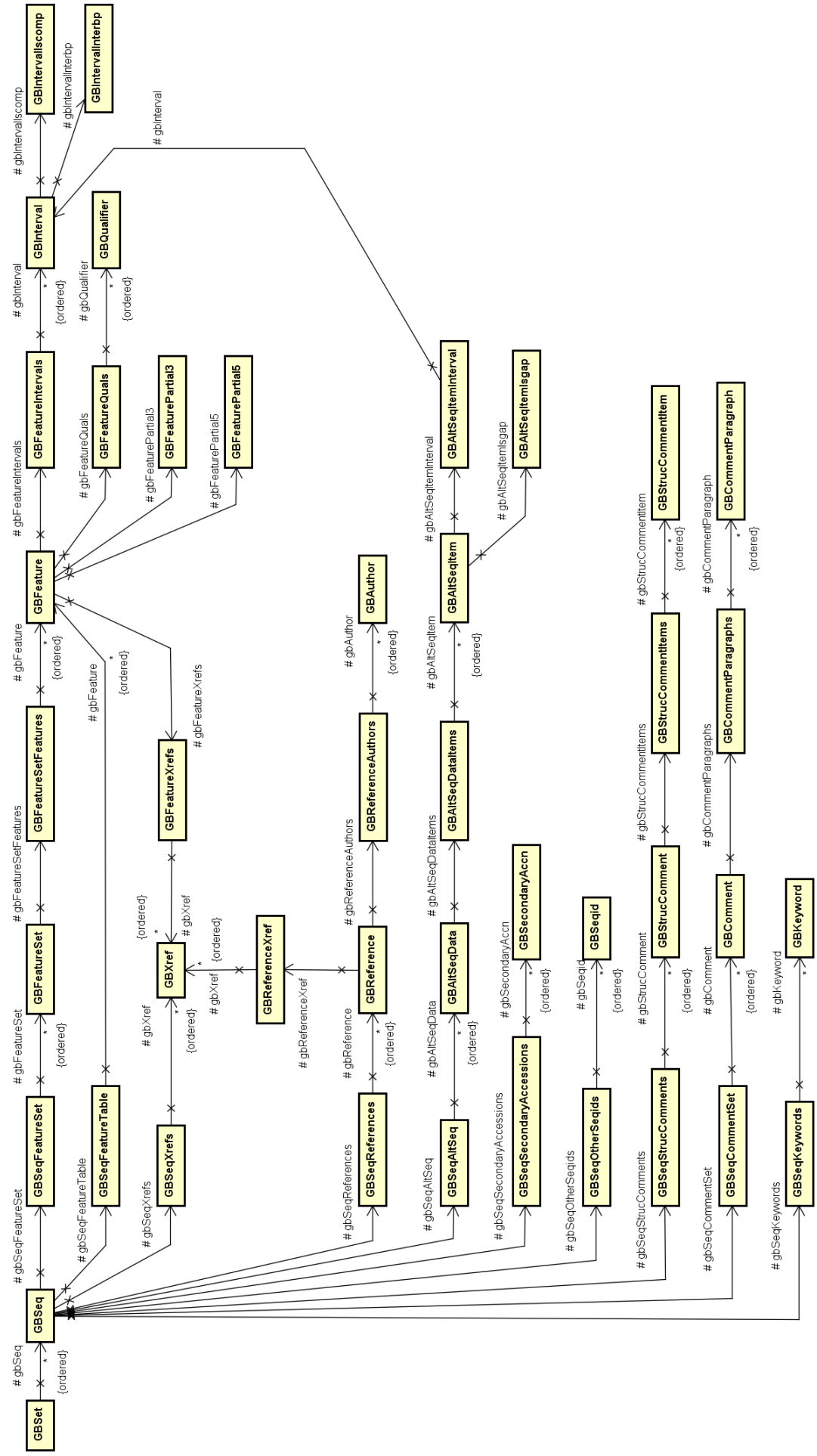
3. **Extração de dados do NCBI:** Fazendo uso do serviço do Entrez, representado pela seta 1.2 da Figura 63, o módulo de extração, usando a lista do TAD “CryToxin” criada no passo anterior, cria uma cadeia de identificadores das proteínas, e submete essa cadeia ao Entrez para que sejam obtidos os dados de cada uma delas. A *Uniform Resource Locator* (URL) <https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi> é utilizada para acessar o Entrez, sendo necessário transmitir alguns parâmetros na mesma:

   - **tool:** o nome da ferramenta que está acessando o Entrez. Nesse caso, “crygetter”;
   - **email:** o e-mail do responsável pela ferramenta. Nesse caso, “davidbuzatto@ifsp.edu.br”;
   - **db:** o banco de dados que será acessado. Nesse caso, “protein”;
   - **retmode:** qual a forma de serialização que será usada pelo serviço para responder à requisição. Nesse caso, “xml”;
   - **id:** a cadeia de identificadores, obrigatoriamente separados por vírgulas, que identificam cada uma das proteínas desejadas. Nesse caso, a cadeia gerada no passo anterior do módulo de extração;
   - ***Obs:** outros detalhes em reação à interface *Hypertext Transfer Protocol Secure* (HTTPS) de acesso ao Entrez podem ser encontrados na sua documentação: MLA... (2016).

   O retorno gerado pelo Entrez é então processado pelo módulo de extração, gerando um conjunto de arquivos temporários no sistema de arquivos que está executando a

ferramenta, sendo que os mesmos serão usados na última etapa de extração. Os dados desses arquivos serão deserializados em uma composição de objetos que permanecerá armazenada em memória principal posteriormente. O processo de deserialização nesse cenário será apoiado pela composição de classes apresentada no diagrama de classes mostrado na Figura 66. Nesse diagrama os compartimentos de atributos e de operações das classes foram escondidos, visando apresentar um diagrama mais sucinto.

Figura 66 – Diagrama de classes que representam o XML retornado pelo Entrez



Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

4. **Geração do arquivo de extração:** O último passo da execução do módulo de extração é relacionado à geração de um arquivo de extração que será armazenado pelo usuário da ferramenta. A gravação desse arquivo é representada pela seta 1.3 da Figura 63, sendo que o mesmo conterá todos os dados que foram obtidos nos passos anteriores e poderá ser aberto por qualquer usuário que tenha a ferramenta.

O processo de carga ou abertura de um arquivo de extração é feito clicando-se no botão "*Load*", destacado na seção A2, em azul, na Figura 64, sendo que, ao ser clicado, um diálogo de abertura de arquivo será mostrado ao usuário, que escolherá o arquivo desejado para ser aberto. Quando esse processo é executado, o "Módulo de Carga" do CryGetter, destacado em laranja e pelo número 2 na Figura 63, entra em atividade, descompactando o arquivo de extração escolhido (seta 2.1 da Figura 63) e apresentando seu conteúdo na "Interface Gráfica com o Usuário", destacada em amarelo e pelo número 3 na Figura 63, além de alguns dados como data de extração e quantidade de proteínas processadas, que serão exibidos nas caixas de texto destacadas na seção A5, em amarelo, da Figura 64.

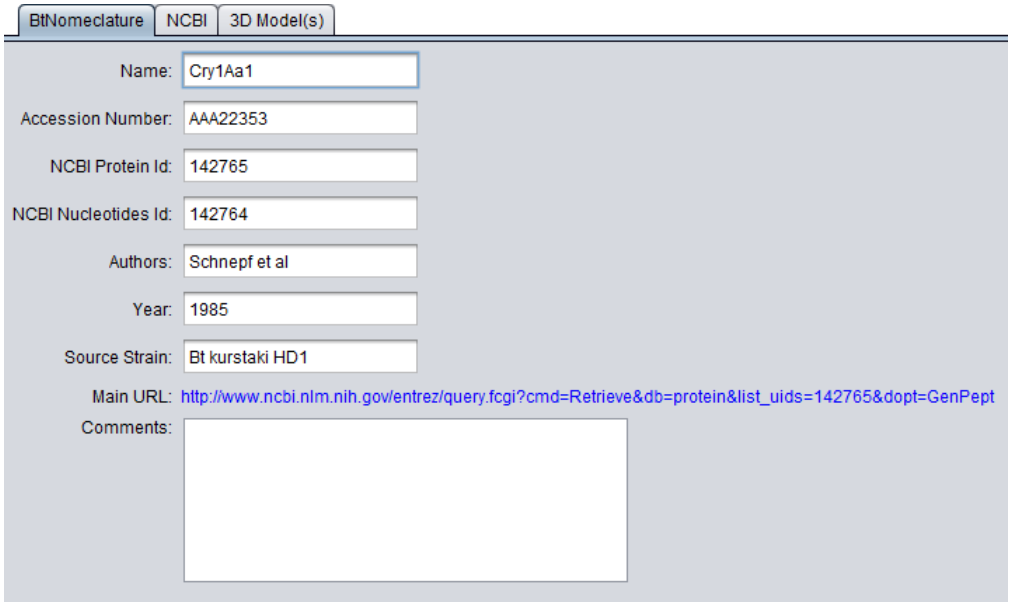
A apresentação dos dados na interface gráfica é a mesma apresentada na seção B da Figura 64, ou seja, todos os dados das proteínas Cry obtidas são carregados na ferramenta, preenchendo a lista destacada na seção A3, em verde, da Figura 64, permitindo que o usuário da ferramenta selecione a proteína que desejar exibir os dados, sendo que os mesmos serão apresentados nas abas destacadas na seção B1, em vermelho, da Figura 64. Além disso, ao usuário será permitida: 1) a execução dos algoritmos de alinhamento de sequências, função acessada por meio do botão "*Alignment*", destacado pela seção A3, em roxo, da Figura 64; 2) a visualização e análise dos alinhamentos, e; 3) a geração de arquivos FASTA com as sequências de todas as proteínas, função essa acessada ao se clicar no botão "FASTA Gen." destacado na seção A6, em rosa, da Figura 64. A maioria das tarefas que o usuário pode executar são gerenciadas por uma ou mais bibliotecas e/ou ferramentas de terceiros, representadas pela seção "Ferramentas e Bibliotecas de Terceiros" que está colorida em verde e identificada pelo número 4 na Figura 63. Essas tarefas serão detalhadas a seguir.

5.3 VISUALIZAÇÃO DE DADOS DAS PROTEÍNAS CRY

A funcionalidade principal da ferramenta está relacionada à visualização dos dados das proteínas Cry. Para que o usuário possa visualizar esses dados, o mesmo necessita selecionar uma proteína na lista de proteínas (seção A4 da Figura 64) que é preenchida após a carga do arquivo de extração. Ao selecionar uma proteína, por exemplo, a Cry1Aa1, todos os dados dessa proteína, que foram obtidos na extração, são apresentados na interface gráfica, mais especificamente no conjunto de abas mostrado na seção B1 da Figura 64, sendo que as mesmas estão organizadas de forma hierárquica:

- **BtNomenclature:** Nessa aba são apresentados os dados que foram coletados no site *Bt Nomenclature*, sendo que a mesma é apresentada na Figura 67. É possível notar que nessa aba os campos de texto estão organizados de forma similar aos dados catalogados no site *Bt Nomenclature*, ou seja, seguindo as colunas especificadas na tabela apresentada na Figura 61;

Figura 67 – Detalhes dos dados importados no site *Bt Nomenclature*

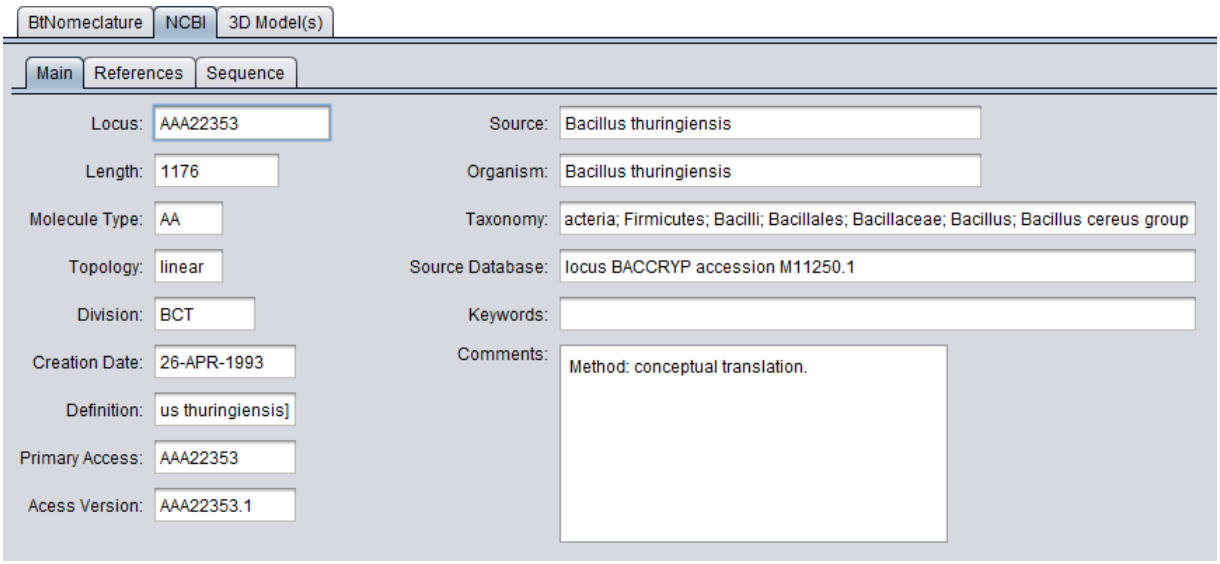


Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

- **NCBI:** Na aba NCBI são apresentados todos os dados que foram obtidos por meio da consulta ao NCBI pelo serviço Entrez durante a extração de dados. Essa aba é dividida em três sub-abas:

  - ***Main*:** Nessa sub-aba são apresentados todos os dados contidos no preâmbulo do resultado da consulta, por exemplo, o tamanho total da sequência da proteína, qual o organismo fonte, ou seja, qual organismo a sintetiza, entre outros, sendo que essa aba pode ser vista na Figure 68;

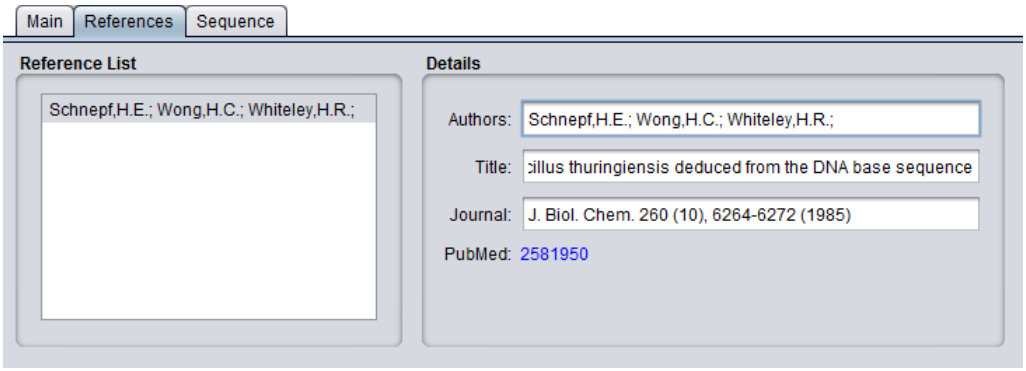
Figura 68 – Detalhes dos dados importados do NCBI - Aba Principal



Fonte: Adaptado de Buzatto, França e Zingaretti (2016) pelo autor

– ***References*:** Na sub-aba *references* são listados o/os trabalhos que originaram o depósito da sequência da proteína. Por exemplo, na Figura 69, pode ser visto que para a proteína Cry1Aa1 há um trabalho, o de Schnepf, Wong e Whiteley (1985);

Figura 69 – Detalhes dos dados importados do NCBI - Aba de Referências

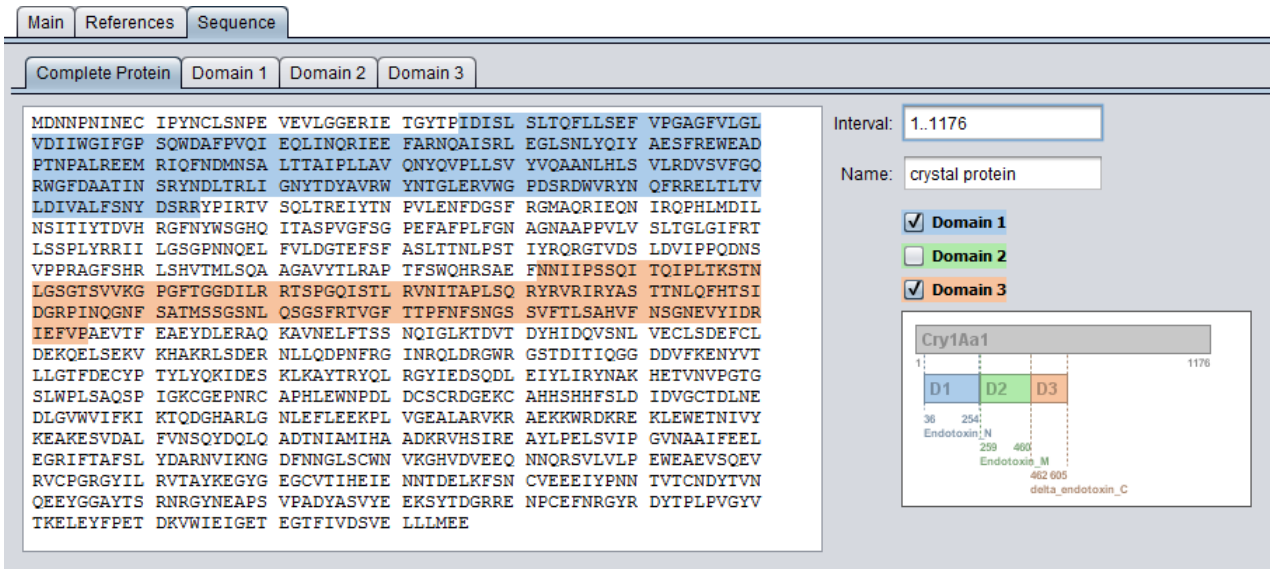


Fonte: Adaptado de Buzatto, França e Zingaretti (2016) pelo autor

– ***Sequence*:** Na última sub-aba da aba NCBI é apresentada a sequência de aminoácidos, ou seja, a estrutura primária da proteína selecionada, bem como os intervalos correspondentes aos três domínios das proteínas Cry, quando existirem. A exibição desses dados é dividida em quatro sub-abas, uma para a sequência completa e uma para cada domínio. Na exibição da sequência completa, apresentada na Figura 70, é exibido, além da sequência, um diagrama que representa a mesma e seus domínios. Além disso, é permitido ao usuário destacar cada um dos domínios ao se clicar nas caixas de seleção correspondentes,

sendo que, na Figura 70, os Domínios I e III estão destacados respectivamente em azul e laranja.

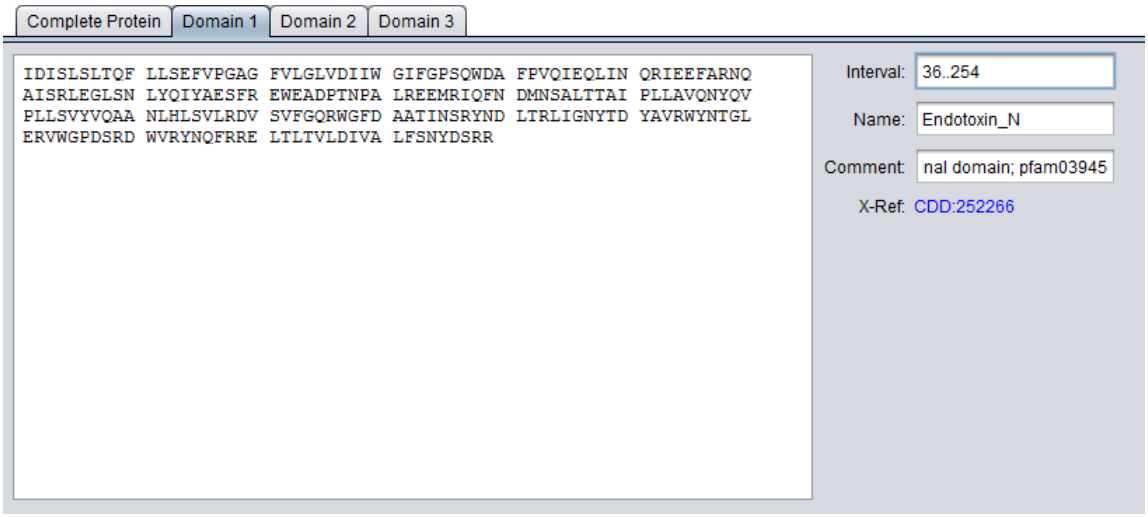
Figura 70 – Detalhes dos dados importados do NCBI - Aba de Sequência



Fonte: Adaptado de Buzatto, França e Zingaretti (2016) pelo autor

A sub-aba que apresenta os dados do intervalo correspondente ao Domínio I é apresentada na Figura 71, permitindo ao usuário visualizar os aminoácidos que fazem parte do intervalo, além de alguns detalhes como o posicionamento do intervalo dentro da sequência completa, o nome dado ao intervalo e algum comentário inserido pelo(s) autore(s) do estudo.

Figura 71 – Detalhes dos dados importados do NCBI - Aba do Domínio I

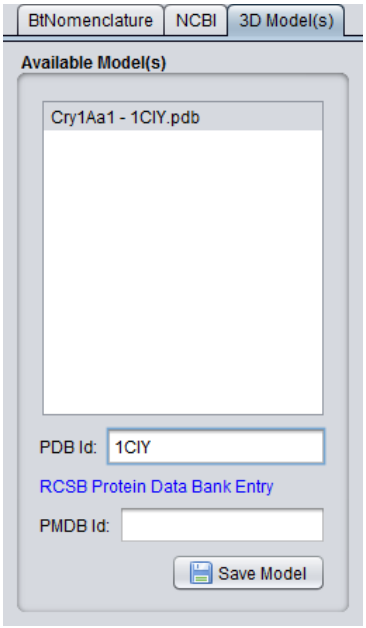


Fonte: Adaptado de Buzatto, França e Zingaretti (2016) pelo autor

- **3D *Model(s)*:** Nessa aba, apresentada na Figura 72, são listados, quando existirem, os modelos tridimensionais da proteína Cry em questão, permitindo que o

usuário salve o modelo desejado e o processe em ferramentas de visualização de proteínas, como por exemplo: VMD (HUMPHREY; DALKE; SCHULTEN, 1996); Jmol (JMOL. . . , 2016); Swiss PDB Viewer (GUEX; PEITSCH, 1997), e; PyMol (SCHRODINGER, 2015).

Figura 72 – Detalhes dos modelos tridimensionais



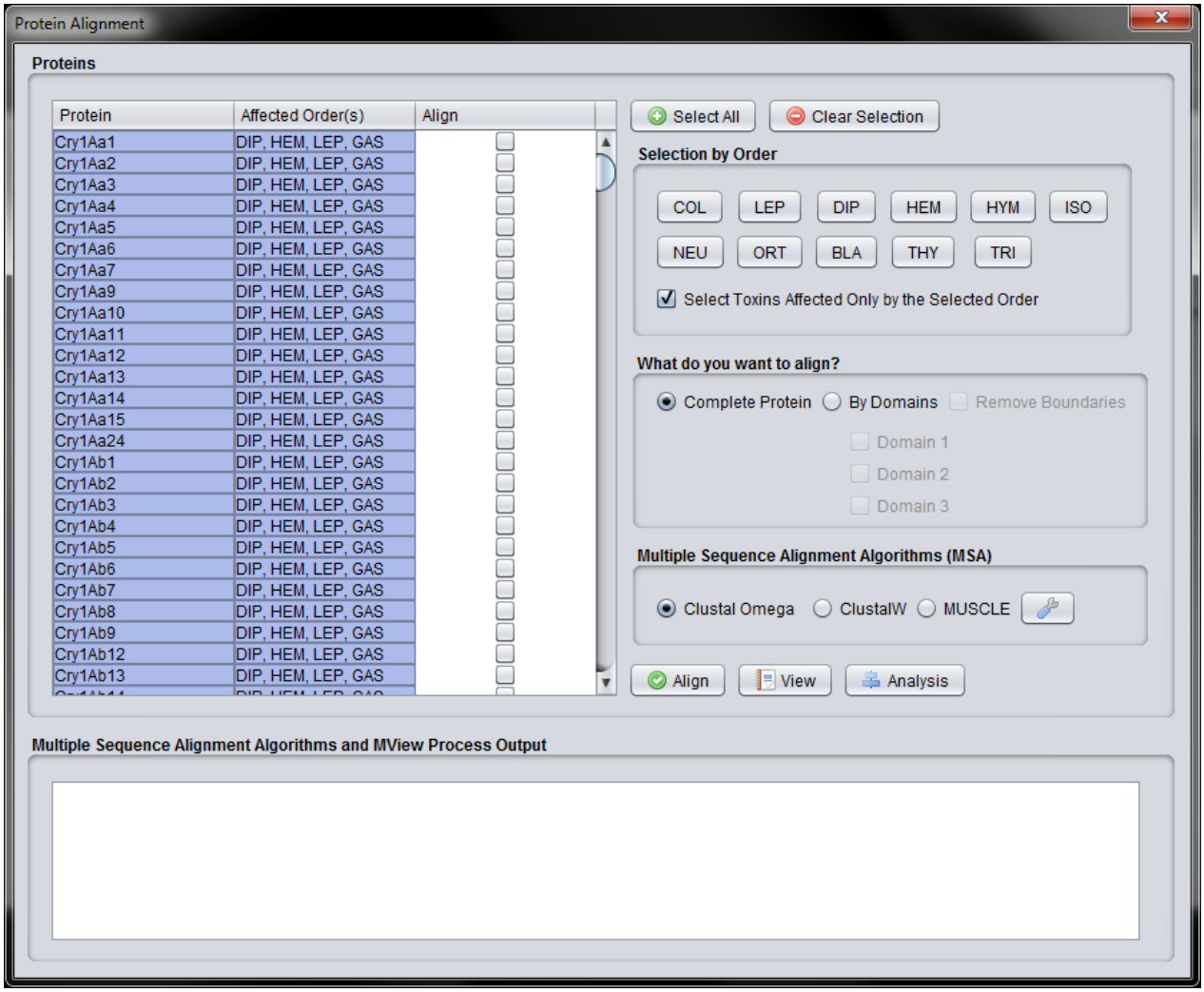
Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

No exemplo apresentado anteriormente, é mostrado que para a proteína Cry1Aa1 existe um modelo tridimensional depositado no PDB e com identificador igual a 1CIY.

5.4 ALINHAMENTO DE PROTEÍNAS CRY

A interface gráfica de alinhamento das proteínas Cry, intitulada "*Cry Protein Alignment*", é acessada ao se clicar no botão "*Alignment*" destacado na seção A3, em roxo, da Figura 64. Essa interface pode ser vista na Figura 73 e é usada, como seu nome indica, para se executar alinhamentos de sequências nas proteínas Cry que o usuário desejar, gerando arquivos de alinhamento como resultado e que são indicados pela seta 4.1 na Figura 63.

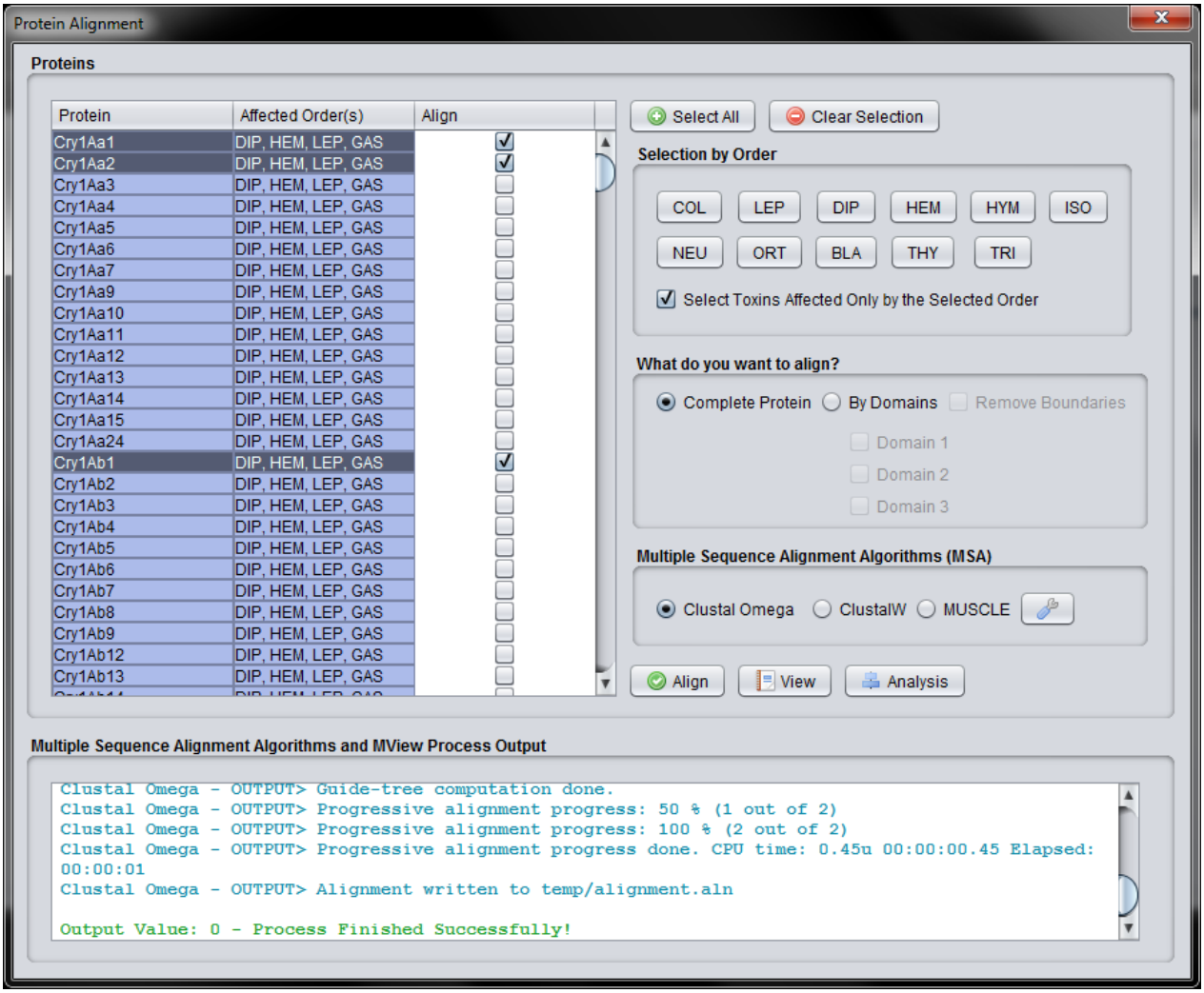
Figura 73 – Interface de alinhamento



Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

Como exemplo, na Figura 74, pode-se ver o estado da interface de alinhamento após a execução do algoritmo Clustal *Omega* nas sequências das proteínas Cry1Aa1, Cry1Aa2 e Cry1Ab1. Quando o algoritmo termina sua execução, um diálogo é exibido, permitindo ao usuário salvar o arquivo de alinhamento (seta 4.1 na Figura 63) que contém os dados do processamento executado. Esse arquivo de alinhamento é usado pela funcionalidade de análise de alinhamentos que será apresentada na próxima Seção.

Figura 74 – Interface de alinhamento após processamento

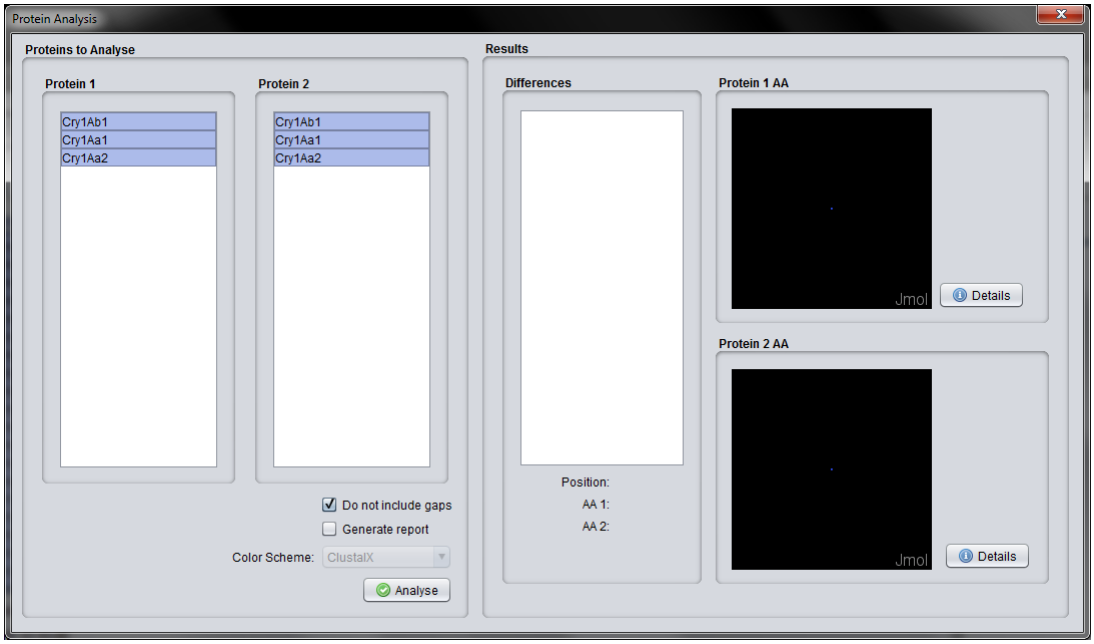


Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

5.5 PROCESSAMENTO DOS RESULTADOS DO ALINHAMENTO

A partir do arquivo de alinhamento de duas ou mais proteínas, o CryGetter é capaz de executar algumas análises preliminares nos resultados gerados. Ao se clicar no botão "*Analysis*", contido na interface de alinhamento (Figura 73), é requisitado ao usuário que escolha o arquivo de alinhamento desejado e, posteriormente, a interface de análise, intitulada "*Protein Analysis*", é exibida. Essa interface pode ser vista na Figura 75.

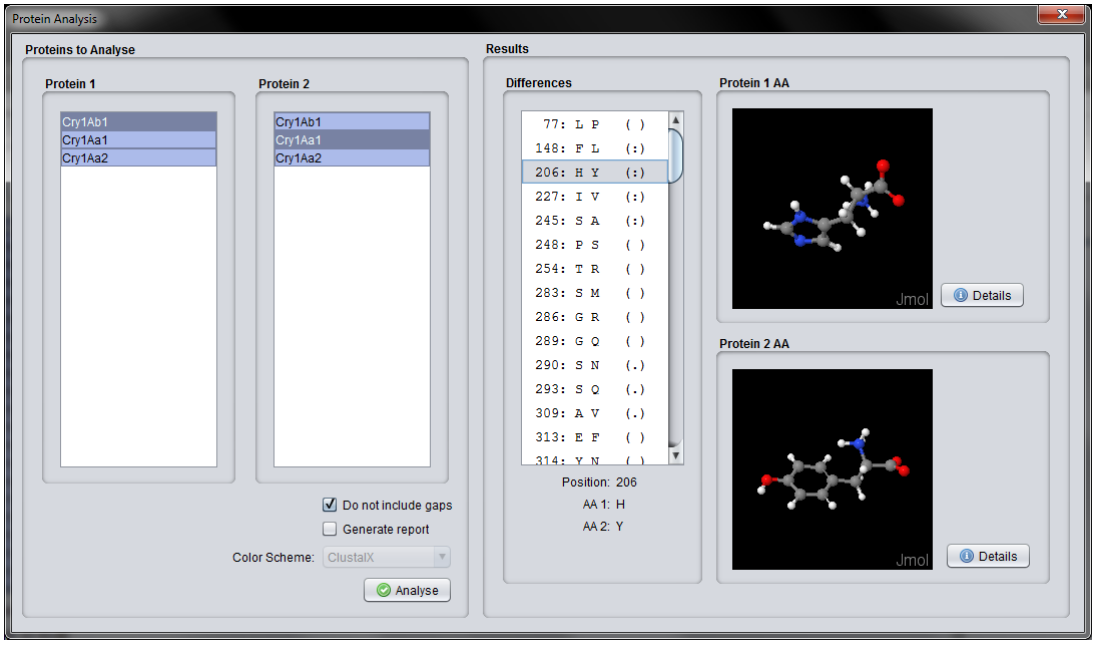
Figura 75 – Interface de análise



Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

Com a interface carregada, o usuário pode então disparar o processo de análise. Para isso, ele deve selecionar um par de proteínas que deseja processar. Após a seleção, o mesmo deve clicar no botão "*Analyse*". No exemplo apresentado na Figura 74 foi optado por analisar as proteínas Cry1Ab1 e Cry1Aa1 e o resultado gerado foi apresentado na lista denominada "*Differences*". Nesse caso, na posição 206, a proteína Cry1Ab1 tem um aminoácido ^{205}H^{207} (Histidina) enquanto a proteína Cry1Aa1 tem um aminoácido ^{205}Y^{207} (Tirosina). Ao usuário também é permitida a geração de um relatório com alguns resultados. Essa funcionalidade será apresentada na próxima Seção, em que será mostrado um exemplo completo do uso da ferramenta.

Figura 76 – Interface de análise após processamento



Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

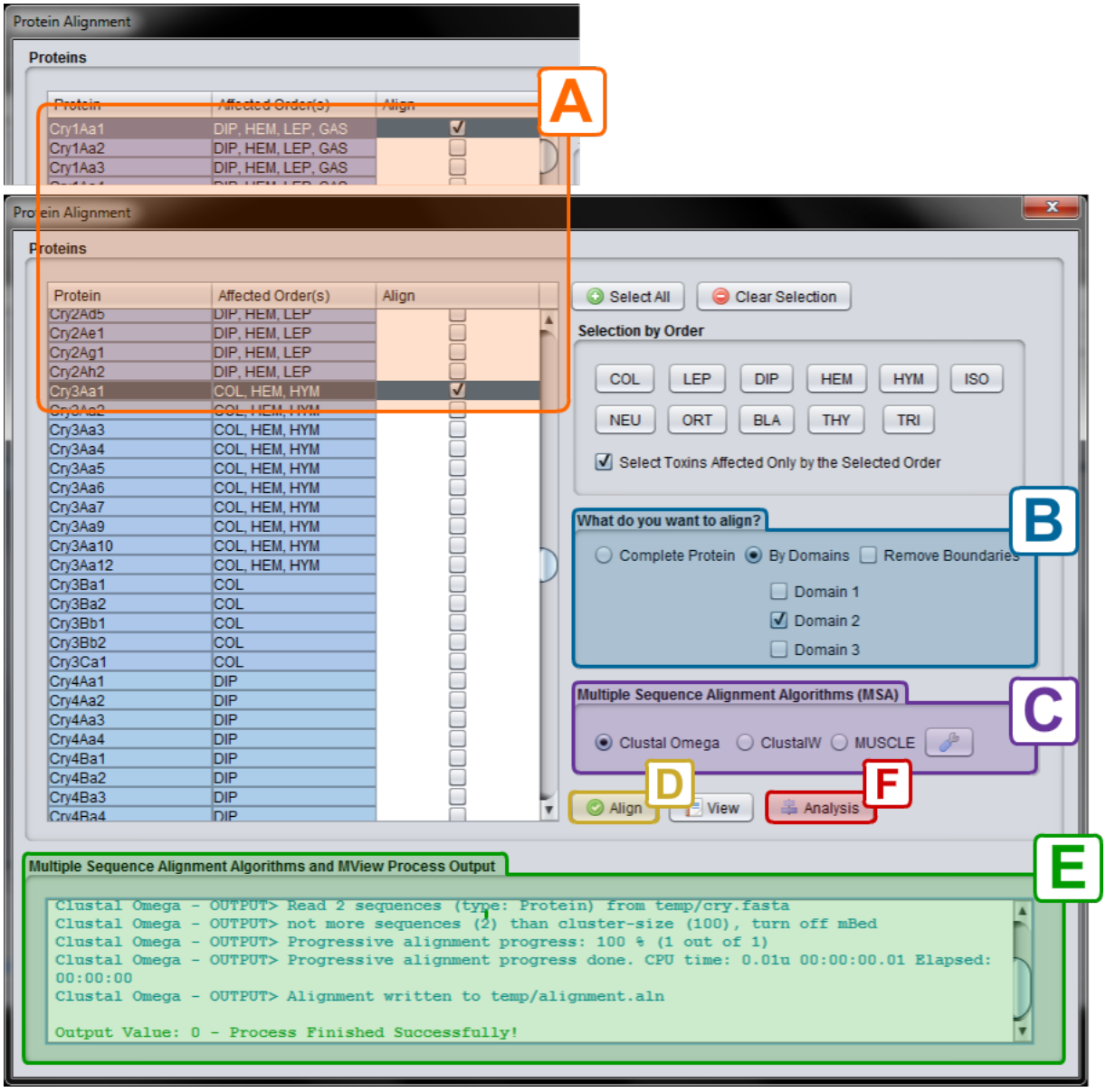
5.6 EXEMPLO DE USO DO CRYGETTER

Como forma de facilitar o entendimento sobre o uso da ferramenta, a seguir será apresentada uma situação em que um usuário deseja processar duas proteínas Cry para que sejam gerados dados preliminares para algum estudo futuro. As proteínas que serão processadas são:

- **Cry1Aa1:** essa proteína afeta principalmente a ordem *Diptera*;
- **Cry3Aa1:** essa proteína afeta principalmente a ordem *Coleoptera*.

Com um arquivo de extração já gerado, o usuário primeiramente o carrega na ferramenta. Após a carga, o usuário terá acesso à interface de alinhamento ao clicar no botão "*Alignment*", destacado na seção A3, em roxo, da Figura 64. Com a interface de alinhamento aberta, são selecionadas então as proteínas desejadas, no caso, Cry1Aa1 e Cry3Aa1. A seleção dessas duas proteínas é apresentada na seção A, em laranja, da Figura 77.

Figura 77 – Configuração de experimento de exemplo usando o CryGetter



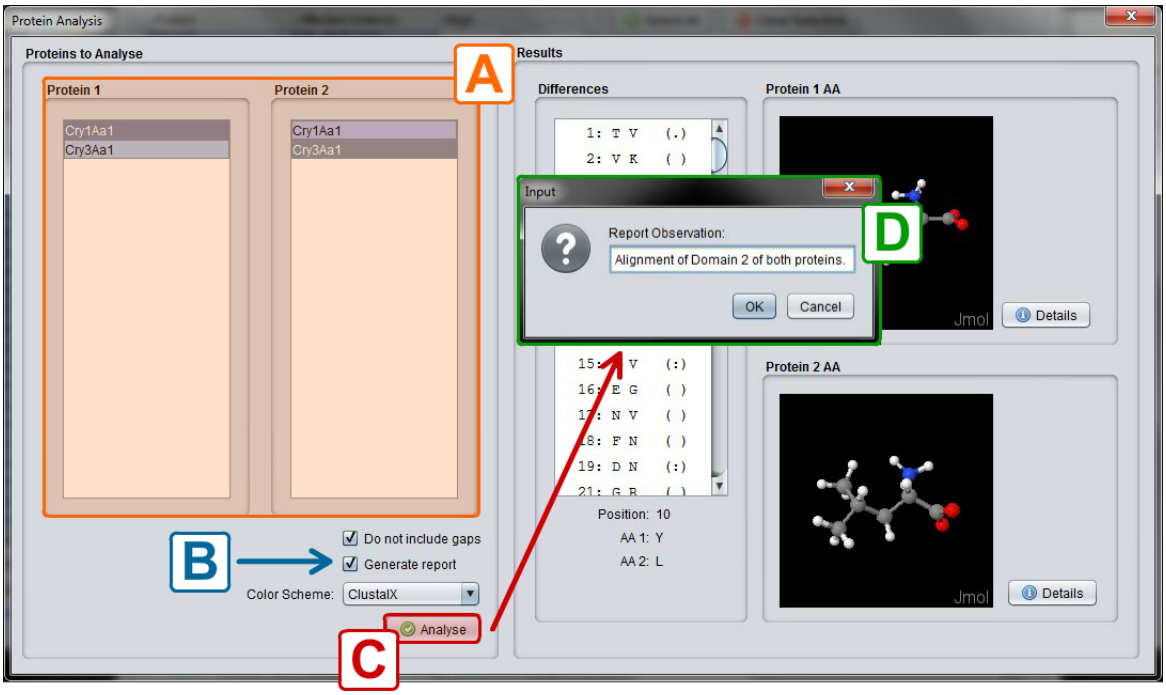
Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

Antes de executar o alinhamento, o usuário deve selecionar na seção B, em azul, da Figura 77, qual trecho das duas proteínas ele deseja alinhar. Nesse caso foi selecionado o Domínio II, visto que o mesmo, segundo Maagd, Bravo e Crickmore (2001), é responsável pelo reconhecimento do receptor das células que serão infectadas e o usuário está interessado em encontrar evidências na estrutura primária que possam corroborar com essa afirmação. Na seção C, em roxo, da Figura 77 é então selecionado o algoritmo de alinhamento que será utilizado. Nesse caso, o usuário optou por utilizar o algoritmo Clustal *Omega*. Finalmente, após a configuração do que deseja fazer, o usuário clica no botão "*Align*", destacado em amarelo na seção D da Figura 77. Ao fazer isso, o algoritmo Clustal *Omega* é executado sobre os dados que foram configurados, sendo que o resultado do processo é exibido na seção E, em verde, da Figura 77. Nesse caso, tudo ocorreu normalmente, visto que a última

linha do resultado apresenta a mensagem "*Process Finished Successfully!*", fazendo com que a ferramenta dê a opção ao usuário de salvar o alinhamento executado.

Com o alinhamento salvo, o usuário pode então processá-lo para que sejam gerados os resultados preliminares que o mesmo procura. Para isso, o botão "*Analysis*", destacado na seção F em vermelho da Figura 77, deve ser clicado, fazendo com que a ferramenta requisite ao usuário o arquivo de alinhamento que deve ser processado. Ao selecionar o arquivo desejado, a interface de análise é apresentada, permitindo sua configuração. Nesse caso, como apresentado na Figura 78, inicia-se o processo de configuração, em que foram selecionadas as proteínas Cry1Aa1 como a primeira proteína e a proteína Cry3Aa1 como a segunda. Essa seleção é feita nas listas destacadas na seção A, em laranja, da Figura 78. O usuário então opta pela criação do relatório de resultados ao clicar na caixa de seleção "*Generate report*", indicada pela letra B, em azul, na Figura 78. Com a configuração de análise pronta, o usuário clica, por fim, no botão "*Analyse*", destacado na seção C, em vermelho, da Figura 78. Ao fazer isso, é apresentado um diálogo para inserção de observações. Nesse caso, o usuário optou em informar que o relatório apresenta dados relativos ao alinhamento do Domínio II das proteínas. Ao clicar no botão "OK" do diálogo apresentado, o relatório de resultados é preparado e exibido.

Figura 78 – Análise do experimento de exemplo usando o CryGetter



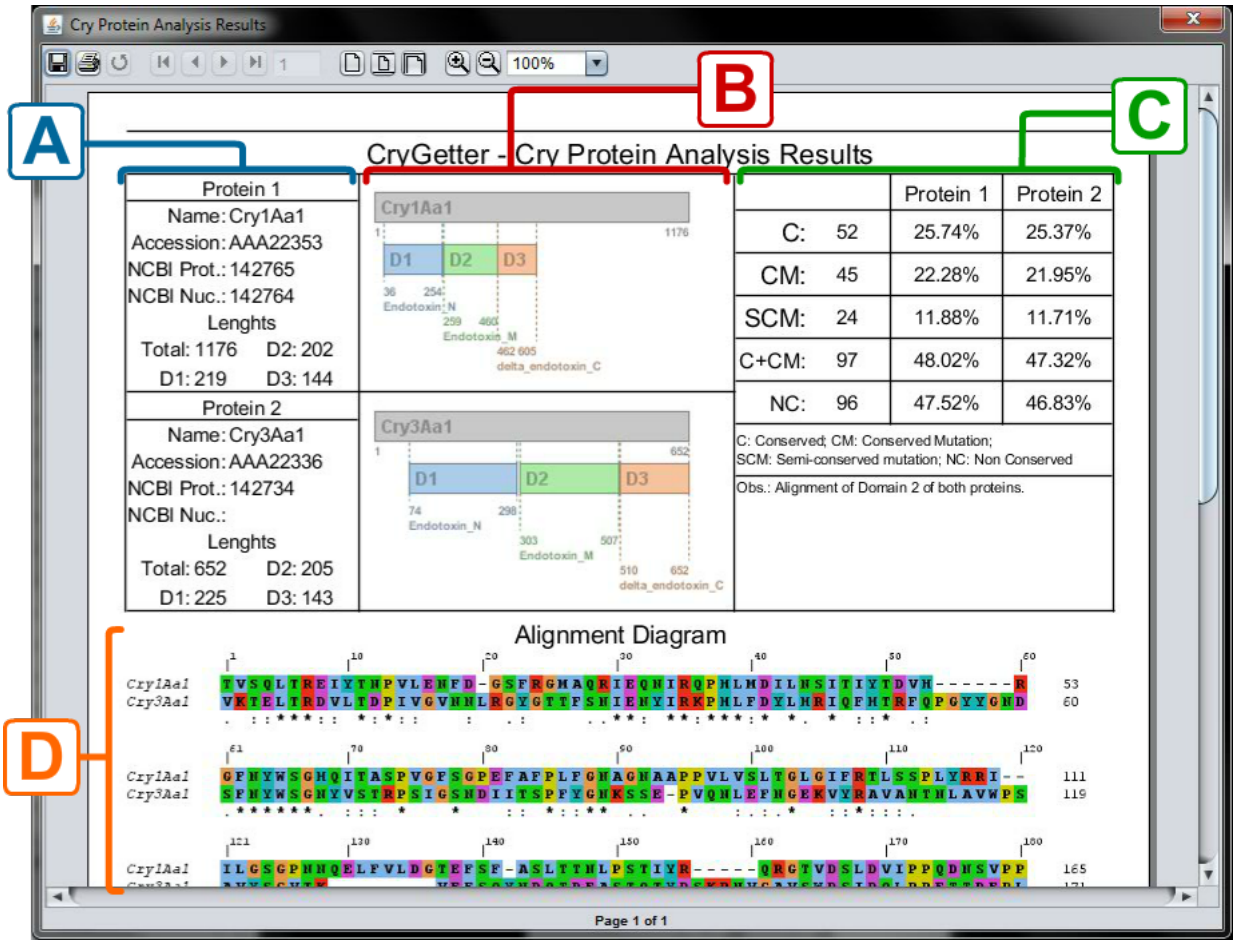
Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

No relatório que é gerado, apresentado na Figura 79, é apresentado um sumário de ambas as proteínas, indicado pela letra A em azul, contendo o nome das proteínas, o *accession number* das entradas no NCBI, os identificadores das sequências de aminoácidos e de nucleotídios, também do NCBI, e um conjunto de valores numéricos que representam os

comprimentos das proteínas completas e de cada um de seus domínios. Na seção indicada pela letra B, em vermelho, da Figura 79 pode-se ver a apresentação, no relatório, dos diagramas das estruturas lineares de ambas as proteínas e, na seção indicada pela letra C, em verde, são apresentadas algumas estatísticas do alinhamento executado, sendo que:

- **C:** representa a quantidade de resíduos conservados;
- **CM:** indica a quantidade de mutações conservadas (*conserved mutations*);
- **SCM:** contém a quantidade de mutações semi-conservadas (*semi-conserved mutations*);
- **C+CM:** é a soma de C e CM;
- **NC:** representa a quantidade de resíduos não conservados.

Figura 79 – Relatório do experimento de exemplo usando o CryGetter



Fonte: Reproduzida na íntegra de Buzatto, França e Zingaretti (2016) pelo autor

As porcentagens que são apresentadas nas colunas "*Protein* 1" e "*Protein* 2" são relacionadas ao valor de cada item em relação ao total de resíduos utilizados no alinhamento,

e não obrigatoriamente ao tamanho total das sequências. Nesse exemplo, C é igual a 52, que no caso corresponde a 25,74% dos 202 resíduos do Domínio II da proteína 1 e 25,37% de 205 resíduos do Domínio II da proteína 2.

Por fim, o diagrama completo do alinhamento é exibido na parte de baixo do relatório, indicada no exemplo pela letra D em laranja. Como já explanado, a análise executada pela ferramenta pode ser usada como ponto inicial na comparação de duas ou mais proteínas Cry.

Sendo assim, com a ferramenta pronta, foi possível entender melhor a natureza das proteínas Cry, além de contribuir com a comunidade de pesquisadores que estudam essas proteínas, fornecendo uma ferramenta capaz de realizar análises preliminares nas mesmas. Essa contribuição se deu pela publicação do artigo "BUZATTO, D.; FRANÇA, S. de C.; ZINGARETTI, S. M. CryGetter: a tool to automate retrieval and analysis of Cry protein data. *BMC Bioinformatics*, v. 17, n. 1, p. 1–14, 2016.", além da criação de uma página na Internet para divulgação da ferramenta, bem como a disponibilização do seu código fonte completo, e que pode ser acessada pelo endereço <http://davidbuzatto.github.io/CryGetter/>.

O próximo passo na metologia deste trabalho foi a obtenção dos modelos tridimensionais que foram depositados em bancos de dados de estruturas de proteínas, visto que, o objetivo principal desta pesquisa, é comparar a estrutura das proteínas Cry para que se possa verificar se há relação entre a conformação dessas proteínas com a toxicidade manifestada nos insetos das ordens afetadas. Na próxima Seção será apresentado como esses modelos foram encontrados, além de como se deu a escolha dos modelos que serão processados nos experimentos.

5.7 MODELOS TRIDIMENSIONAIS

Para a comparação das estruturas tridimensionais das proteínas Cry, inicialmente foi avaliada a possibilidade de se gerar esses modelos usando ferramentas computacionais para a predição estrutural das mesmas, o que, em teoria, permitiria a comparação estrutural de quaisquer proteínas que se quisesse verificar, entretanto, a utilização desses tipos de ferramentas gera modelos não confiáveis, visto que em sua maioria, a predição estrutural é feita a partir de modelos já existentes e que apresentam homologia com a estrutura primária das proteínas que se deseja modelar. Sendo assim, a melhor opção para a obtenção dos modelos tridimensionais seria fazer uso dos bancos de dados de modelos de proteínas, no caso o PDB e do PMDB.

Ao pesquisar esses bancos de dados e na literatura sobre os modelos tridimensionais das proteínas Cry que foram obtidos experimentalmente, obteve-se os dados apresentados na Tabela 5, estes publicados por Buzatto, França e Zingaretti (2016), sendo que a

partir desses dados, pode-se verificar três situações importantes: 1) modelos depositados e publicados; 2) modelos depositados, mas com publicação pendente, e; 3) modelos não depositados e com publicação.

Sendo assim, os dados da Tabela 5, que totalizam 25 proteínas distintas com 27 modelos, foram filtrados, gerando a Tabela 8.

Tabela 8 – Modelos das proteínas Cry depositados

Proteína	**Ordem(s) Afetadas**	**Identificador do Modelo**	**Referência(s)**	**Observações**
Cry1Aa1	*Diptera*, *Lepidoptera* e *Gastropoda*	1CIY	Knowles e Ellar (1987) e Grochulski et al. (1995)	
Cry1Ac1	*Diptera*, *Lepidoptera* e *Gastropoda*	4ARX 4ARY 4W8J	Ainda não publicado Ainda não publicado Derbyshire, Ellar e Li (2001)	
Cry2Aa1	*Diptera*, *Hemiptera* e *Lepidoptera*	1I5P	Morse, Yamamoto e Stroud (2001)	
Cry3A	*Coleoptera*, *Hemiptera* e *Hymenoptera*	1DLC	Li, Carroll e Ellar (1991)	
Cry3Aa1	*Coleoptera*, *Hemiptera* e *Hymenoptera*	4QX0 4QX1 4QX2 4QX3	Sawaya et al. (2014)	
Cry3Bb1	*Coleoptera*	1JI6	Galitsky et al. (2001)	
Cry4Aa1	*Diptera*	2C9K	Boonserm, Angsuthanasombat e Lescar (2004) e Boonserm et al. (2006)	
Cry4Ba1	*Diptera*	1W99 4MOA	Boonserm et al. (2005) Sriwimol et al. (2015)	
Cry5Aa1	*Hymenoptera* e *Rhabditida*	PM0074964	Xin-Min et al. (2009)	
Cry5B	*Rhabditida*	4D8M	Hui et al. (2012)	
Cry5Ba1	*Rhabditida*	PM0075036	Xia et al. (2008)	
Cry6Aa	*Rhabditida*	5J66 5KUC 5KUD	Dementiev et al. (2016)	Formação de poro
Cry6Aa2	*Rhabditida*	5GHE	Huang et al. (2016)	Formação de poro
Cry8Ea1	*Coleoptera*	3EB7	Guo et al. (2009)	
Cry23Aa1 Cry37Aa1	*Coleoptera*	4RHZ	Ainda não publicado	Complexo proteico binário
Cry34Ab1	*Coleoptera*	4JOX	Kelker et al. (2014)	
Cry35Ab1	*Coleoptera*	4JP0	Kelker et al. (2014)	
Cry51Aa1	*Coleoptera* e *Hemiptera*	4PKM	Xu et al. (2015)	
Cry51Aa2	*Coleoptera* e *Hemiptera*	5HD2	Gowda et al. (2016)	

* Os identificadores de modelo que possuem códigos com quatro caracteres representam identificadores do PDB, enquanto os identificadores com 9 caracteres são códigos do PMDB.

Fonte: Adaptado de Buzatto, França e Zingaretti (2016) pelo autor

A filtragem feita consistiu em remover todos os modelos que foram citados na literatura, mas que não foram disponibilizados eletronicamente, no caso os modelos das proteínas Cry1Ab16, Cry1Ab19, Cry1Ld, Cry11Bb1 e Cry30Ca2. Entrou-se em contato com os autores das publicações que citavam os supostos modelos que foram gerados, requisitando os mesmos, entretanto, não foram obtidas respostas. Sendo assim, na Tabela 8, são listadas 20 proteínas com 27 modelos. A quantidade de modelos é maior que a quantidade de

proteínas, pois há proteínas, por exemplo, a Cry1Ac1, que possuem mais de um modelo.

O próximo passo na filtragem dos modelos foi remover todos os casos de proteínas Cry que não possuem três domínios ou que no modelo os três domínios não estão mapeados, além de excluir modelos do PMDB caso existam modelos da mesma proteína no PDB e que são ativas contra a mesma ordem. Com essa filtragem foi obtida a Tabela 9.

Tabela 9 – Modelos das proteínas Cry depositados que possuem três domínios

Proteína	**Ordem(s) Afetadas**	**Identificador do Modelo**
Cry1Aa1	*Diptera*, *Lepidoptera* e *Gastropoda*	1CIY
Cry1Ac1	*Diptera*, *Lepidoptera* e *Gastropoda*	4ARX 4ARY 4W8J
Cry2Aa1	*Diptera*, *Hemiptera* e *Lepidoptera*	1I5P
Cry3A	*Coleoptera*, *Hemiptera* e *Hymenoptera*	1DLC
Cry3Aa1	*Coleoptera*, *Hemiptera* e *Hymenoptera*	4QX0 4QX1 4QX2 4QX3
Cry3Bb1	*Coleoptera*	1JI6
Cry4Aa1	*Diptera*	2C9K
Cry4Ba1	*Diptera*	1W99 4MOA
Cry5B	*Rhabditida*	4D8M
Cry8Ea1	*Coleoptera*	3EB7

Fonte: Adaptado de Buzatto, França e Zingaretti (2016) pelo autor

Para obter a Tabela 9, foram removidas as proteínas Cry6Aa, Cry6Aa2, Cry23Aa1, Cry37Aa1, Cry34Ab1, Cry35Ab1, Cry51Aa1 e Cry51Aa2, que são proteínas Cry que não possuem três domínios ou não têm os três domínios mapeados no modelo tridimensional, e as proteínas Cry5Aa1 e Cry5Ba1, ambas depositadas no PMDB, que tem ação contra a ordem *Rhabditida* do filo *Nematoda*. Essas últimas duas proteínas foram removidas, pois será dada preferência para a proteína Cry5B que possui depósito no PDB e que também é ativa contra a mesma ordem. Obteve-se, assim, 10 proteínas com 15 modelos. Como há proteínas com mais de um modelo, o último passo da filtragem dos modelos que serão usados nos experimentos é remover os modelos duplicados que possuem métricas de qualidade piores que os outros. No caso, foram analisados os modelos de identificadores PDB 4ARX, 4ARY e 4W8J da proteína Cry1Ac1; 1DLC, 4QX0, 4QX1, 4QX2 e 4QX3

da proteína Cry3A/Cry3Aa1, e; 1W99 e 4MOA da proteína Cry4Ba1, sendo que esses modelos, com suas respectivas métricas, podem ser vistos na Tabela 10.

Tabela 10 – Métricas de qualidade dos modelos das proteínas Cry

Proteína	**Identificador do Modelo**	**Resolução**	**R_{free}**	**Clashscore**	**Ramachandran *Outliers***	***Sidechain Outliers***	**RSRZ *Outliers***
Cry1Ac1	4ARX	2,35Å	0,221	2	0,0%	0,6%	2,8%
	4ARY	2,95Å	0,244	3	0,0%	0,5%	2,0%
	4W8J	2,78Å	0,322	11	0,5%	6,4%	4,5%
Cry3A e Cry3Aa1	1DLC	2,50Å	-	9	0,3%	4,7%	-
	4QX0	2,80Å	0,215	5	0,3%	4,5%	0,0%
	4QX1	2,80Å	0,196	4	0,3%	5,1%	0,2%
	4QX2	2,90Å	0,227	5	0,5%	4,3%	0,3%
	4QX3	2,90Å	0,201	5	0,3%	5,3%	0,3%
Cry4Ba1	1W99	1,75Å	0,220	7	0,0%	1,0%	6,8%
	4MOA	2,00Å	0,223	18	2,0%	3,7%	13,3%

Fonte: Relatórios de validação estrutural de cada modelo, disponíveis no PDB

Em relação às métricas apresentadas, o significado de cada uma delas pode ser encontrado no manual do usuário dos relatórios de validação do PDB (WWPDB. . . , 2016):

- **Resolução:** Nível de detalhe do modelo. Quanto menor, melhor;
- **R_{free}:** Mede quanto o modelo se ajusta a um subconjunto de dados experimentais. Quanto menor, maior o ajuste;
- **Clashscore:** Derivado da quantidade de pares de átomos que estão próximos de forma incomum. Quanto menor, melhor;
- **Ramachandran *Outliers*:** Mede a quantidade de ângulos de torção ϕ e ψ que possuem valores incomuns, ou seja, que estão fora das regiões permitidas no gráfico de Ramachandran. Quanto menor, melhor;
- ***Sidechain Outliers*:** Mede a quantidade de cadeias laterais que estão dispostas de forma incomum. Quanto menor, melhor;
- **RSRZ (*R-value Z-score*) *Outliers*:** É a porcentagem de resíduos não ajustados ao modelo atômico de dados no espaço real em relação aos resíduos ajustados. Quanto menor, melhor.

Sendo assim, para que haja o desempate dos modelos de uma mesma proteína, escolheu-se os modelos que possuem os melhores valores de cada métrica: Cry1Ac1, modelo 4ARX; Cry3A/Cry3Aa1, modelo 4QX0, e; Cry4Ba1, modelo 1W99. Com isso, foram escolhidos os modelos que serão utilizados nos experimentos de comparação estrutural. Esses modelos estão apresentados na Tabela 11.

Tabela 11 – Modelos das proteínas Cry escolhidos

Proteína	**Ordem(s) Afetadas**	**Identificador do Modelo**
Cry1Aa1	*Diptera*, *Lepidoptera* e *Gastropoda*	1CIY
Cry1Ac1	*Diptera*, *Lepidoptera* e *Gastropoda*	4ARX
Cry2Aa1	*Diptera*, *Hemiptera* e *Lepidoptera*	1I5P
Cry3Aa1	*Coleoptera*, *Hemiptera* e *Hymenoptera*	4QX0
Cry3Bb1	*Coleoptera*	1JI6
Cry4Aa1	*Diptera*	2C9K
Cry4Ba1	*Diptera*	1W99
Cry5B	*Rhabditida*	4D8M
Cry8Ea1	*Coleoptera*	3EB7

Fonte: Adaptado de Buzatto, França e Zingaretti (2016) pelo autor

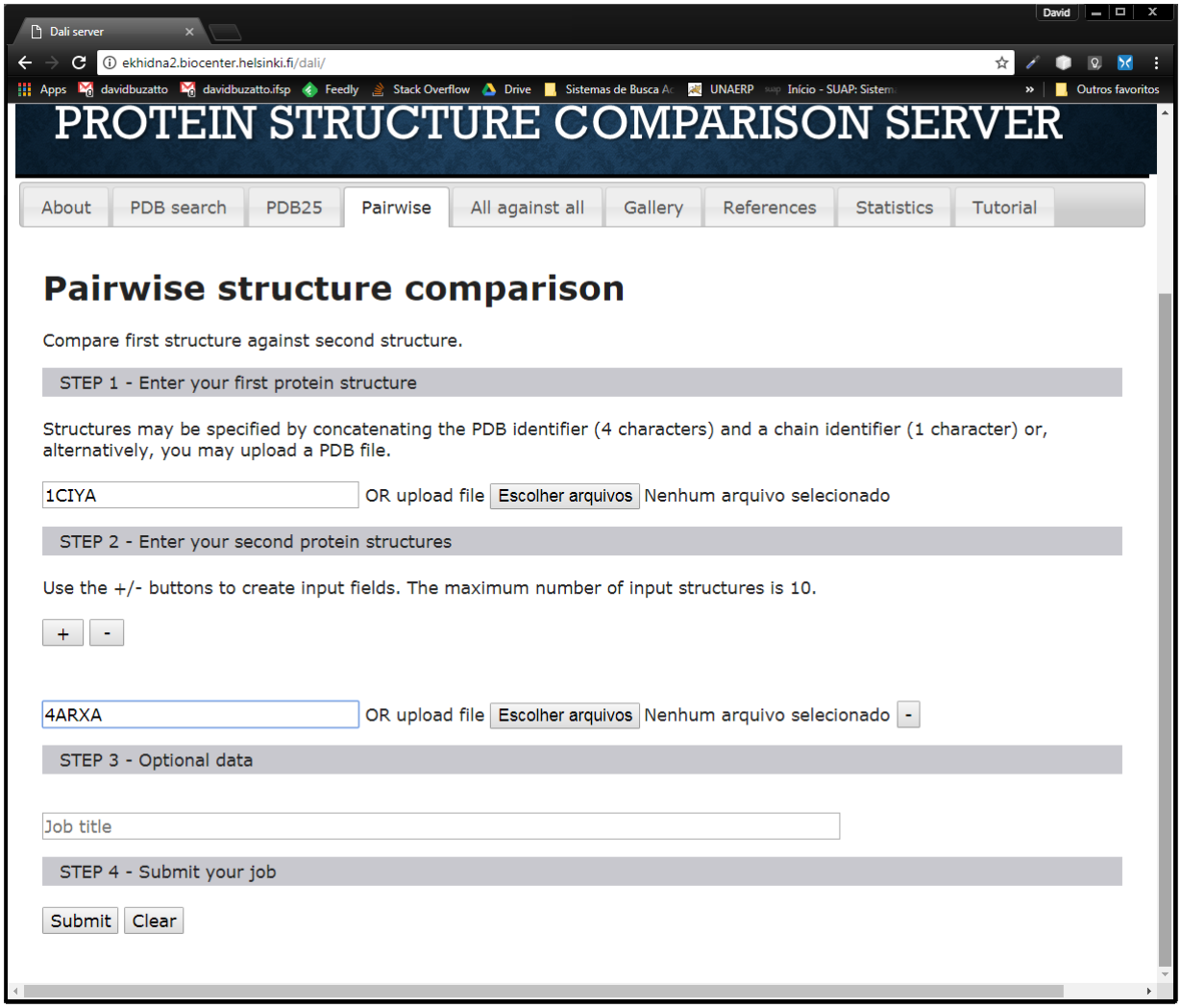
Na Tabela 11 são apresentadas as 9 proteínas que serão analisadas, com seus respectivos modelos, sendo que os mesmos serão usados nos experimentos de comparação estrutural, pois além de não haver duplicação de modelos, pode-se também notar que há variabilidade nas ordens afetadas pelas proteínas representadas pelos mesmos. A forma com que esses experimentos foram conduzidos será apresentada na próxima Seção.

5.8 EXPERIMENTOS

Para que a comparação das estruturas das proteínas listadas na Tabela 11 fosse executada, foram escolhidos primeiramente três algoritmos de alinhamento estrutural apresentados na revisão da literatura: Dali, FatCat e CE.

O algoritmo Dali foi escolhido por ser uma ferramenta que tem sido desenvolvida e melhorada há mais de 20 anos, além de ser uma citação recorrente em diversos trabalhos sobre alinhamento estrutural de proteínas. A execução do Dali se deu pela função de alinhamento de pares, que pode ser acessada pelo endereço <http://ekhidna2.biocenter.helsinki.fi/dali/> e é mostrada na interface gráfica apresentada na Figura 80.

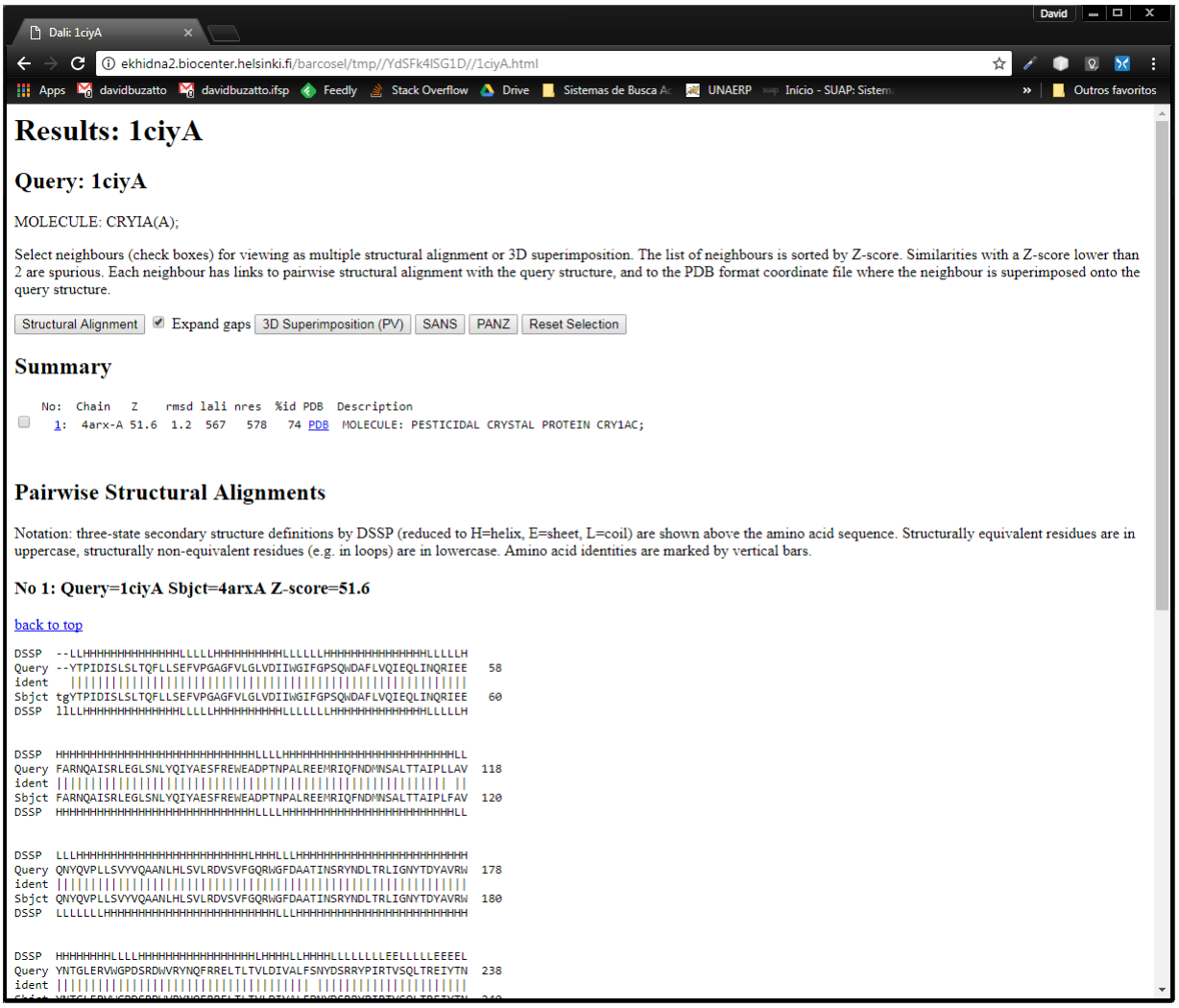
Figura 80 – Tela principal de execução do algoritmo Dali para pares de estruturas



Fonte: Captura de tela. Disponível em: <http://ekhidna2.biocenter.helsinki.fi/dali/>. Acesso em: 10/09/2017

Na Figura 80, pode-se ver que para executar o alinhamento estrutural, é necessário primeiramente fornecer os arquivos ou os identificadores PDB dos modelos, além da cadeia que se quer utilizar para o alinhamento. Nos experimentos deste trabalho sempre foram usadas as cadeias A, visto que em modelos que contém mais de uma cadeia, essas cadeias extra são cópias da cadeia A, mas dispostas de forma diferente no espaço tridimensional. Após fornecer esses dados, que para o exemplo apresentado foram os identificadores PDB 1CIY, da proteína Cry1Aa1, e 4ARX, da proteína Cry1Ac1, deve-se clicar no botão "*Submit*" e aguardar que o alinhamento seja feito. Na Figura 81 pode ser observado o sumário do resultado para esse alinhamento, sendo que nesse sumário, vários alinhamentos possíveis entre os modelos definidos podem ser gerados, sendo esses apresentados em ordem decrescente de qualidade, ou seja, o primeiro alinhamento foi o melhor, o segundo foi pior que o primeiro e assim por diante.

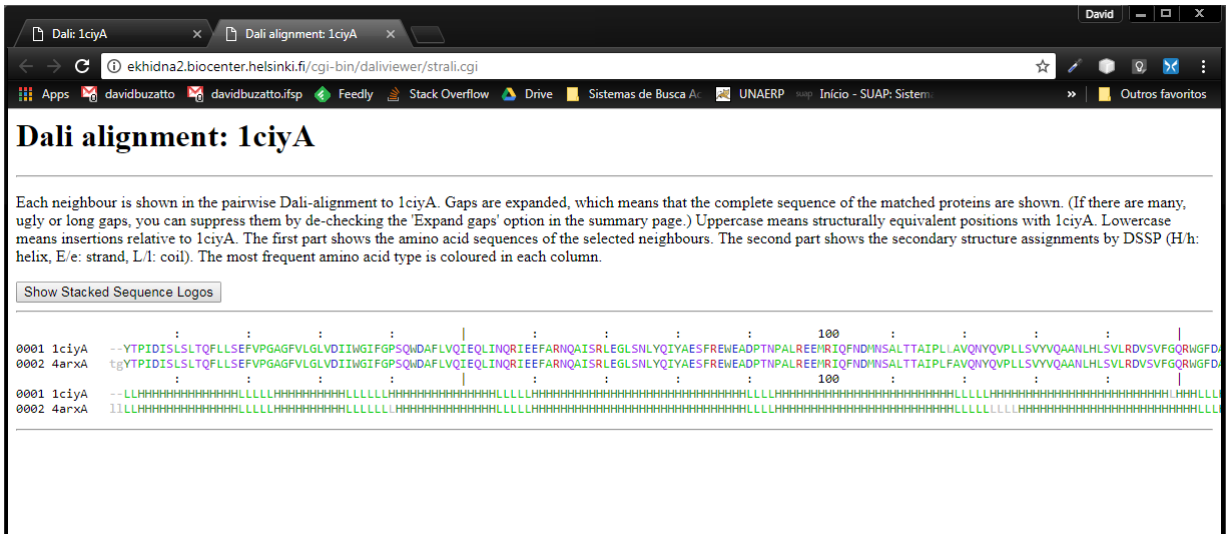
Figura 81 – Resultado do alinhamento das estruturas das proteínas Cry1Aa1 (PDB: 1CIY) e Cry1Ac1 (PDB: 4ARX) (Dali)



Fonte: Captura de tela do resultado do alinhamento estrutural

Para verificar o alinhamento estrutural que foi gerado em um alinhamento específico, deve-se selecionar o alinhamento desejado no sumário, clicando na caixa de seleção correspondente, e clicar no botão "*Structural Alignment*". Fazendo isso, é mostrado o alinhamento em questão. Como exemplo, foi selecionado o alinhamento número 1 da lista do sumário apresentado na Figura 81, que nesse caso só possui um alinhamento disponível, clicou-se no botão "*Structural Alignment*" e foi obtido o resultado desse alinhamento, cuja representação gráfica pode ser observada na Figura 82.

Figura 82 – Resultado do melhor alinhamento das estruturas das proteínas Cry1Aa1 (PDB: 1CIY) e Cry1Ac1 (PDB: 4ARX) (Dali)



Fonte: Captura de tela do melhor resultado do alinhamento estrutural

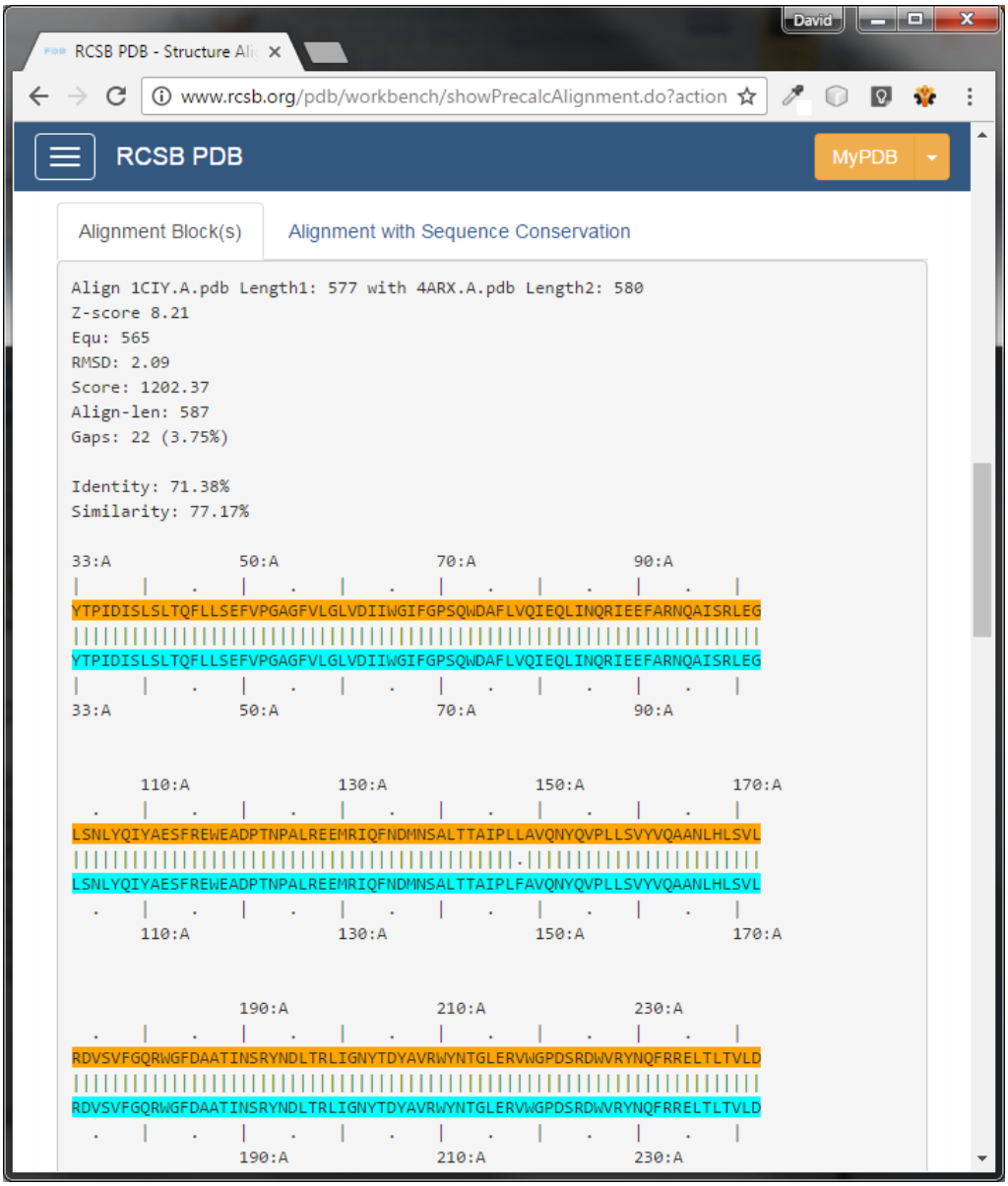
Os outros dois algoritmos, FatCat e CE, são disponibilizados como opções para o alinhamento estrutural no site do PDB, e por isso foram escolhidos também, permitindo que os resultados dos mesmos fossem confrontados com os do algoritmo Dali. Na Figura 83 pode ser vista a interface gráfica para alinhamento estrutural do site do PDB. Nessa interface, assim como na do Dali, fornece-se os identificadores PDB das estruturas desejadas e qual a cadeia de cada modelo se quer alinhar, além de ser necessário definir qual algoritmo deve ser utilizado.

Figura 83 – Tela principal de execução dos algoritmos CE e FatCat no PDB para pares de estruturas

Fonte: Captura de tela. Disponível em: <http://www.rcsb.org/pdb/secondary.do?p=v2/secondary/analyze.jsp#Sequence>. Acesso em: 10/09/2017

No exemplo apresentado na Figura 83 foram alinhados os mesmos modelos do exemplo do algoritmo Dali, sendo que, nesse caso, o algoritmo selecionado para a execução do alinhamento foi o CE, escolha esta mostrada na caixa de seleção com o item "jCE *algorithm*". Após a configuração, deve-se clicar no botão "*Align*" para o alinhamento ser executado. O resultado para esse alinhamento pode ser visto na Figura 84.

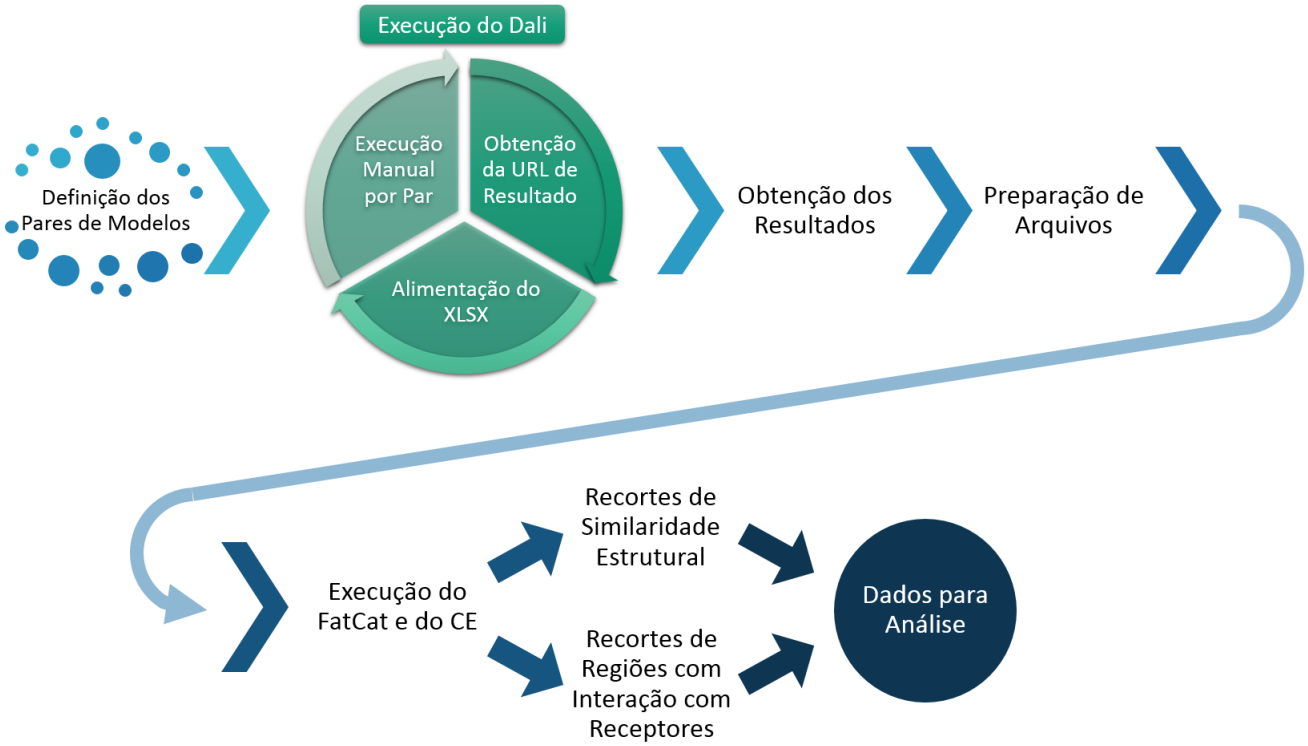
Figura 84 – Resultado do alinhamento das estruturas das proteínas Cry1Aa1 (PDB: 1CIY) e Cry1Ac1 (PDB: 4ARX) (CE)



Fonte: Captura de tela do resultado do alinhamento estrutural

Apesar desses dois algoritmos serem disponibilizados no site do PDB, neste trabalho optou-se por usar a implementação local dos mesmos, disponibilizada pela biblioteca BioJava, desenvolvida por Prlic et al. (2012), permitindo uma semi-automatização do processo de alinhamento e obtenção de resultados. No diagrama apresentado na Figura 85 todo o processo dos experimentos, ou seja, dos alinhamentos dos pares de modelos, é ilustrado.

Figura 85 – Processo de Execução dos Experimentos



Fonte: Elaborada pelo autor

A seguir os passos do processo de execução dos experimentos serão detalhados:

1. **Definição dos Pares de Modelos:** A partir dos nove modelos escolhidos e apresentados na Tabela 11, foram feitas combinações entre eles, sem repetição, gerando a Tabela 12, que no caso contém todos os 36 pares de modelos que foram utilizados para o alinhamento estrutural;

Tabela 12 – Pares de Modelos para o Experimento

Proteína 1	**Modelo 1**	**Proteína 2**	**Modelo 2**
Cry1Aa1	1CIY	Cry1Ac1	4ARX
Cry1Aa1	1CIY	Cry2Aa1	1I5P
Cry1Aa1	1CIY	Cry3Aa1	4QX0
Cry1Aa1	1CIY	Cry3Bb1	1JI6
Cry1Aa1	1CIY	Cry4Aa1	2C9K
Cry1Aa1	1CIY	Cry4Ba1	1W99
Cry1Aa1	1CIY	Cry5B	4D8M
Cry1Aa1	1CIY	Cry8Ea1	3EB7
Cry1Ac1	4ARX	Cry2Aa1	1I5P
Cry1Ac1	4ARX	Cry3Aa1	4QX0
Cry1Ac1	4ARX	Cry3Bb1	1JI6
Cry1Ac1	4ARX	Cry4Aa1	2C9K
Cry1Ac1	4ARX	Cry4Ba1	1W99
Cry1Ac1	4ARX	Cry5B	4D8M
Cry1Ac1	4ARX	Cry8Ea1	3EB7
Cry2Aa1	1I5P	Cry3Aa1	4QX0
Cry2Aa1	1I5P	Cry3Bb1	1JI6
Cry2Aa1	1I5P	Cry4Aa1	2C9K
Cry2Aa1	1I5P	Cry4Ba1	1W99
Cry2Aa1	1I5P	Cry5B	4D8M
Cry2Aa1	1I5P	Cry8Ea1	3EB7
Cry3Aa1	4QX0	Cry3Bb1	1JI6
Cry3Aa1	4QX0	Cry4Aa1	2C9K
Cry3Aa1	4QX0	Cry4Ba1	1W99
Cry3Aa1	4QX0	Cry5B	4D8M
Cry3Aa1	4QX0	Cry8Ea1	3EB7
Cry3Bb1	1JI6	Cry4Aa1	2C9K
Cry3Bb1	1JI6	Cry4Ba1	1W99
Cry3Bb1	1JI6	Cry5B	4D8M
Cry3Bb1	1JI6	Cry8Ea1	3EB7
Cry4Aa1	2C9K	Cry4Ba1	1W99
Cry4Aa1	2C9K	Cry5B	4D8M
Cry4Aa1	2C9K	Cry8Ea1	3EB7
Cry4Ba1	1W99	Cry5B	4D8M
Cry4Ba1	1W99	Cry8Ea1	3EB7
Cry5B	4D8M	Cry8Ea1	3EB7

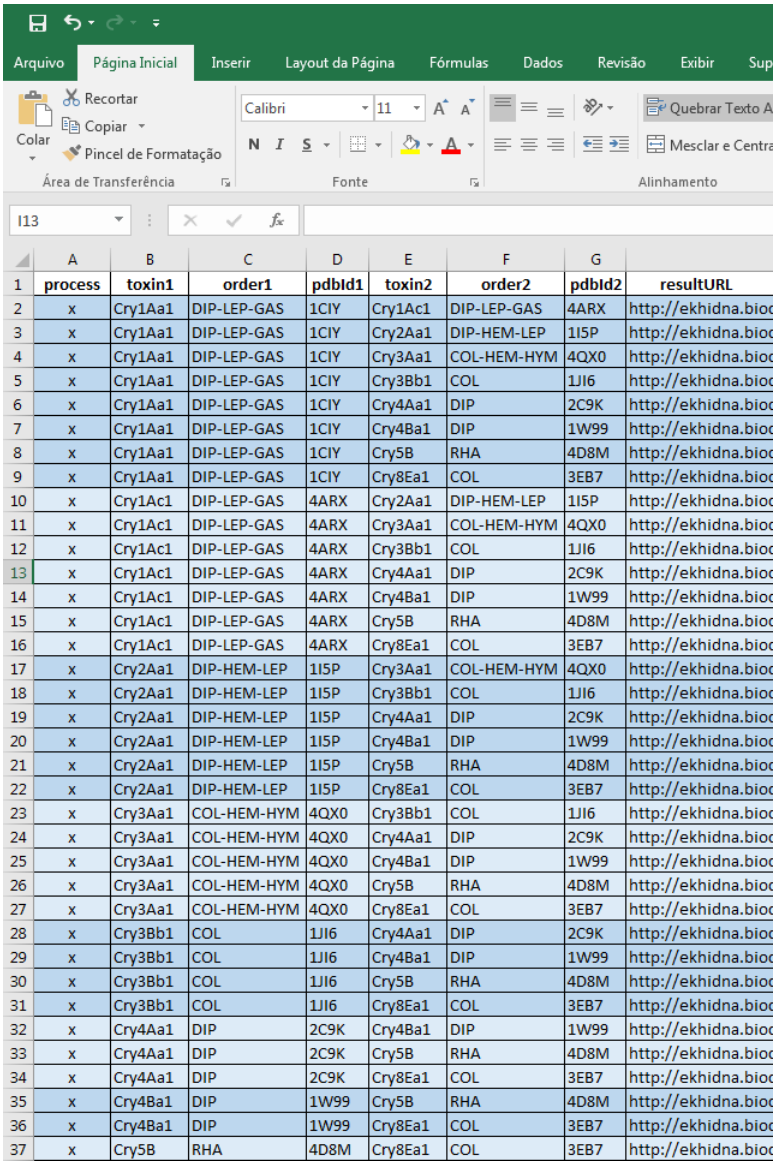
Fonte: Elaborada pelo autor

Para cada um dos pares apresentados, foi necessário utilizar o algoritmo Dali, na versão apresentada anteriormente;

2. **Execução do Dali:** Ao executar o algoritmo para um determinado par, é gerada uma URL que após ser obtida, é inserida em um arquivo Microsoft Excel *Open* XML *Format Spreadsheet* (XLSX) do Microsoft Excel, que será usado como uma base de

dados para a obtenção de todos os dados de cada alinhamento estrutural executado pelo algoritmo Dali. Uma captura de tela desse arquivo pode ser vista na Figura 86;

Figura 86 – Base de Dados em XLSX dos resultados dos experimentos



	A	B	C	D	E	F	G	
1	process	toxin1	order1	pdbId1	toxin2	order2	pdbId2	resultURL
2	x	Cry1Aa1	DIP-LEP-GAS	1CIY	Cry1Ac1	DIP-LEP-GAS	4ARX	http://ekhidna.bioc
3	x	Cry1Aa1	DIP-LEP-GAS	1CIY	Cry2Aa1	DIP-HEM-LEP	1I5P	http://ekhidna.bioc
4	x	Cry1Aa1	DIP-LEP-GAS	1CIY	Cry3Aa1	COL-HEM-HYM	4QX0	http://ekhidna.bioc
5	x	Cry1Aa1	DIP-LEP-GAS	1CIY	Cry3Bb1	COL	1JI6	http://ekhidna.bioc
6	x	Cry1Aa1	DIP-LEP-GAS	1CIY	Cry4Aa1	DIP	2C9K	http://ekhidna.bioc
7	x	Cry1Aa1	DIP-LEP-GAS	1CIY	Cry4Ba1	DIP	1W99	http://ekhidna.bioc
8	x	Cry1Aa1	DIP-LEP-GAS	1CIY	Cry5B	RHA	4D8M	http://ekhidna.bioc
9	x	Cry1Aa1	DIP-LEP-GAS	1CIY	Cry8Ea1	COL	3EB7	http://ekhidna.bioc
10	x	Cry1Ac1	DIP-LEP-GAS	4ARX	Cry2Aa1	DIP-HEM-LEP	1I5P	http://ekhidna.bioc
11	x	Cry1Ac1	DIP-LEP-GAS	4ARX	Cry3Aa1	COL-HEM-HYM	4QX0	http://ekhidna.bioc
12	x	Cry1Ac1	DIP-LEP-GAS	4ARX	Cry3Bb1	COL	1JI6	http://ekhidna.bioc
13	x	Cry1Ac1	DIP-LEP-GAS	4ARX	Cry4Aa1	DIP	2C9K	http://ekhidna.bioc
14	x	Cry1Ac1	DIP-LEP-GAS	4ARX	Cry4Ba1	DIP	1W99	http://ekhidna.bioc
15	x	Cry1Ac1	DIP-LEP-GAS	4ARX	Cry5B	RHA	4D8M	http://ekhidna.bioc
16	x	Cry1Ac1	DIP-LEP-GAS	4ARX	Cry8Ea1	COL	3EB7	http://ekhidna.bioc
17	x	Cry2Aa1	DIP-HEM-LEP	1I5P	Cry3Aa1	COL-HEM-HYM	4QX0	http://ekhidna.bioc
18	x	Cry2Aa1	DIP-HEM-LEP	1I5P	Cry3Bb1	COL	1JI6	http://ekhidna.bioc
19	x	Cry2Aa1	DIP-HEM-LEP	1I5P	Cry4Aa1	DIP	2C9K	http://ekhidna.bioc
20	x	Cry2Aa1	DIP-HEM-LEP	1I5P	Cry4Ba1	DIP	1W99	http://ekhidna.bioc
21	x	Cry2Aa1	DIP-HEM-LEP	1I5P	Cry5B	RHA	4D8M	http://ekhidna.bioc
22	x	Cry2Aa1	DIP-HEM-LEP	1I5P	Cry8Ea1	COL	3EB7	http://ekhidna.bioc
23	x	Cry3Aa1	COL-HEM-HYM	4QX0	Cry3Bb1	COL	1JI6	http://ekhidna.bioc
24	x	Cry3Aa1	COL-HEM-HYM	4QX0	Cry4Aa1	DIP	2C9K	http://ekhidna.bioc
25	x	Cry3Aa1	COL-HEM-HYM	4QX0	Cry4Ba1	DIP	1W99	http://ekhidna.bioc
26	x	Cry3Aa1	COL-HEM-HYM	4QX0	Cry5B	RHA	4D8M	http://ekhidna.bioc
27	x	Cry3Aa1	COL-HEM-HYM	4QX0	Cry8Ea1	COL	3EB7	http://ekhidna.bioc
28	x	Cry3Bb1	COL	1JI6	Cry4Aa1	DIP	2C9K	http://ekhidna.bioc
29	x	Cry3Bb1	COL	1JI6	Cry4Ba1	DIP	1W99	http://ekhidna.bioc
30	x	Cry3Bb1	COL	1JI6	Cry5B	RHA	4D8M	http://ekhidna.bioc
31	x	Cry3Bb1	COL	1JI6	Cry8Ea1	COL	3EB7	http://ekhidna.bioc
32	x	Cry4Aa1	DIP	2C9K	Cry4Ba1	DIP	1W99	http://ekhidna.bioc
33	x	Cry4Aa1	DIP	2C9K	Cry5B	RHA	4D8M	http://ekhidna.bioc
34	x	Cry4Aa1	DIP	2C9K	Cry8Ea1	COL	3EB7	http://ekhidna.bioc
35	x	Cry4Ba1	DIP	1W99	Cry5B	RHA	4D8M	http://ekhidna.bioc
36	x	Cry4Ba1	DIP	1W99	Cry8Ea1	COL	3EB7	http://ekhidna.bioc
37	x	Cry5B	RHA	4D8M	Cry8Ea1	COL	3EB7	http://ekhidna.bioc

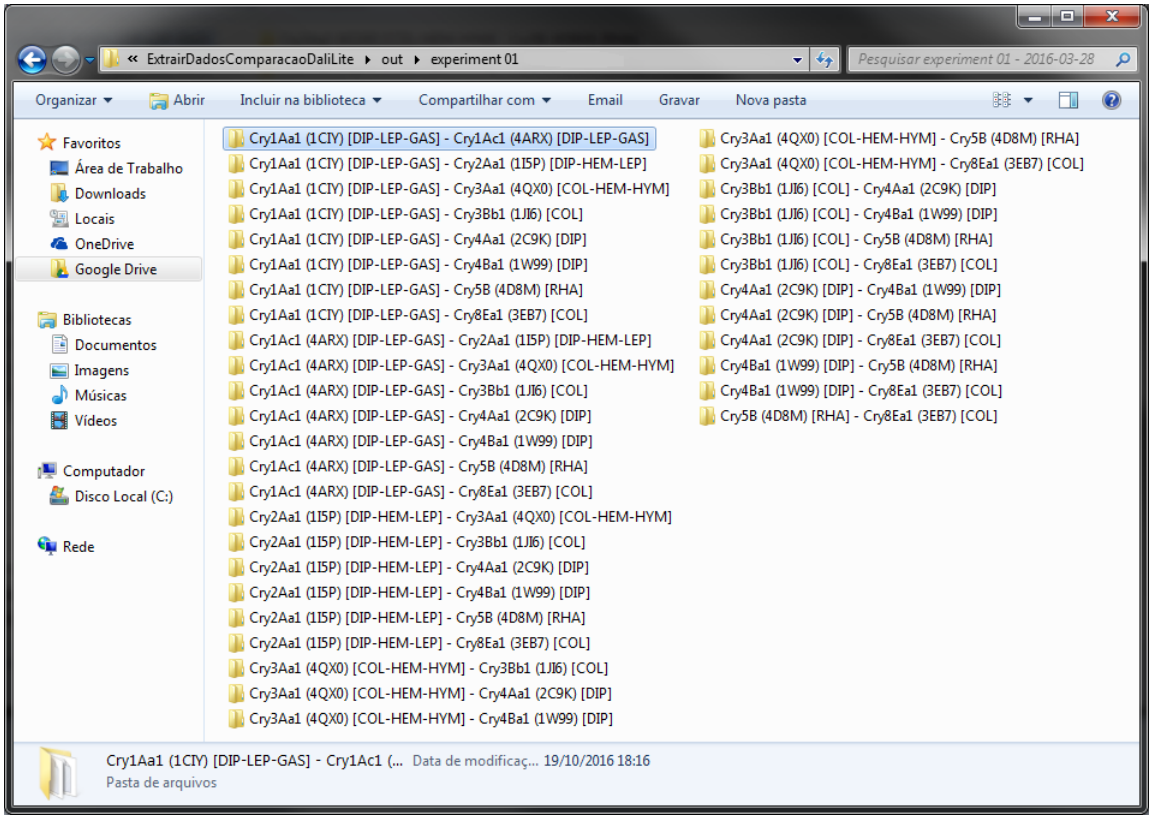
Fonte: Captura de tela do Microsoft Excel

3. **Obtenção dos Resultados:** Esse arquivo XLSX é então processado por um programa que foi desenvolvido neste trabalho, com o objetivo de semi-automatizar a obtenção dos resultados gerados pelo Dali, gerando diversos arquivos e executando também os outros dois algoritmos, bem como de preparar os resultados para análise. Esse programa usa como entrada cada uma das URL obtidas, trazendo os dados dos resultados por meio de uma conexão HTTP;

4. **Preparação de Arquivos:** Após a obtenção dos dados dos resultados de uma URL em particular, o programa gera um conjunto de arquivos, organizados dentro de diretórios específicos. O nome de cada diretório identifica o experimento que foi

executado. Na Figura 87 pode-se observar os 36 diretórios, referentes a cada uma das comparações executadas com base nos pares escolhidos. Nessa mesma figura, um dos diretórios, que está selecionado, indica o experimento do par "Cry1Aa1 (PDB: 1CIY) - Cry1Ac1 (PDB: 4ARX)".

Figura 87 – Diretórios dos Experimentos



Fonte: Captura de tela do Windows Explorer

Os arquivos gerados, como já dito, estão dentro de cada um desses diretórios, sendo eles:

- **alignment-n.txt:** Os resultados do alinhamento n, em texto puro;
- **alignedWith-mol2A-n.pdb:** O segundo modelo rotacionado de acordo com o alinhamento n com o primeiro modelo, no formato .pdb;
- **alignment-jmol-n.spt:** Arquivo de *script* do Jmol para a visualização da superposição dos dois modelos do alinhamento n;
- **alignment-jmol-mol1A-mol2A.pdb:** Modelo com os dois modelos mesclados;
- **index.html:** Cópia da página de resultados, no formato HTML;
- **mol1A-ID.pdb:** Arquivo do primeiro modelo;
- **mol1A-ID.spt:** Arquivo de *script* do Jmol para a visualização do primeiro modelo;

- **mol2A-ID.pdb:** Arquivo do segundo modelo;
- **mol2A-ID.spt:** Arquivo de *script* do Jmol para a visualização do segundo modelo;
- **open-alignment-jmol-n.bat:** Arquivo em lote para a abertura do *script* de visualização do alinhamento n;
- **open-alignment-jmol-mol1A-mol2A.bat:** Arquivo em lote para a abertura do *script* de visualização dos modelos mesclados;
- **open-jmol-mol1A.bat:** Arquivo em lote para a abertura do *script* de visualização do primeiro modelo;
- **open-jmol-mol2A.bat:** Arquivo em lote para a abertura do *script* de visualização do segundo modelo;
- **summary.txt:** Sumário do alinhamento do Dali;
- **run-alignment-ce-cp-mol1A-mol2A.bat:** Arquivo em lotes para a execução do alinhamento dos dois modelos usando o algoritmo CE com permutação circular;
- **run-alignment-ce-mol1A-mol2A.bat:** Arquivo em lotes para a execução do alinhamento dos dois modelos usando o algoritmo CE sem permutação circular;
- **run-alignment-fatcat-flexible-mol1A-mol2A.bat:** Arquivo em lotes para a execução do alinhamento dos dois modelos usando o algoritmo FatCat na variação flexível;
- **run-alignment-fatcat-rigid-mol1A-mol2A.bat:** Arquivo em lotes para a execução do alinhamento dos dois modelos usando o algoritmo FatCat na variação rígida;
- **run-alignment-and-show-ce-cp-mol1A-mol2A.bat:** Arquivo em lotes para a execução e visualização no Jmol do alinhamento dos dois modelos usando o algoritmo CE com permutação circular;
- **run-alignment-and-show-ce-mol1A-mol2A.bat:** Arquivo em lotes para a execução e visualização no Jmol do alinhamento dos dois modelos usando o algoritmo CE sem permutação circular;
- **run-alignment-and-show-fatcat-flexible-mol1A-mol2A.bat:** Arquivo em lotes para a execução e visualização no Jmol do alinhamento dos dois modelos usando o algoritmo FatCat na variação flexível;
- **run-alignment-and-show-fatcat-rigid-mol1A-mol2A.bat:** Arquivo em lotes para a execução e visualização no Jmol do alinhamento dos dois modelos usando o algoritmo FatCat na variação rígida.

Após a geração dos arquivos citados na lista acima, o programa executa os algoritmos FatCat nas variações flexível e rígida e CE com e sem permutação circular;

5. **Execução do FatCat e do CE:** A execução dos algoritmos gera o resultado dos alinhamentos, sendo os mesmos armazenados em mais quatro arquivos:

   - **out-ce-cp.txt:** Resultado do alinhamento dos dois modelos usando o algoritmo CE com permutação circular;
   - **out-ce.txt:** Resultado do alinhamento dos dois modelos usando o algoritmo CE sem permutação circular;
   - **out-fatcat-flexible.txt:** Resultado do alinhamento dos dois modelos usando o algoritmo FatCat na variação flexível;
   - **out-fatcat-rigid.txt:** Resultado do alinhamento dos dois modelos usando o algoritmo FatCat na variação rígida.

   Com os resultados gerados, os mesmos são processados de forma a prepará-los para análise;

6. **Recortes de Similaridade Estrutural e Recortes de Regiões com Interação com Receptores:** Os recortes de similaridade são feitos em cinco arquivos de resultados, sendo eles o do melhor alinhamento do Dali, os dois resultados do FatCat e os dois resultados do CE. Esse recorte gera dados mais fáceis de interpretar, visto que contém apenas os trechos dos alinhamentos estruturais que apresentam algum tipo de similaridade estrutural. O recorte desses cinco arquivos gera mais cinco arquivos:

   - **alignment-1-proc.txt:** Recortes das similaridades encontradas no alinhamento 1, o melhor do Dali;
   - **out-ce-cp-proc.txt:** Recortes das similaridades encontradas no alinhamento usando o algoritmo CE com permutação circular;
   - **out-ce-proc.txt:** Recortes das similaridades encontradas no alinhamento usando o algoritmo CE sem permutação circular;
   - **out-fatcat-flexible-proc.txt:** Recortes das similaridades encontradas no alinhamento usando o algoritmo FatCat na variação flexível;
   - **out-fatcat-rigid-proc.txt:** Recortes das similaridades encontradas no alinhamento usando o algoritmo FatCat na variação rígida;

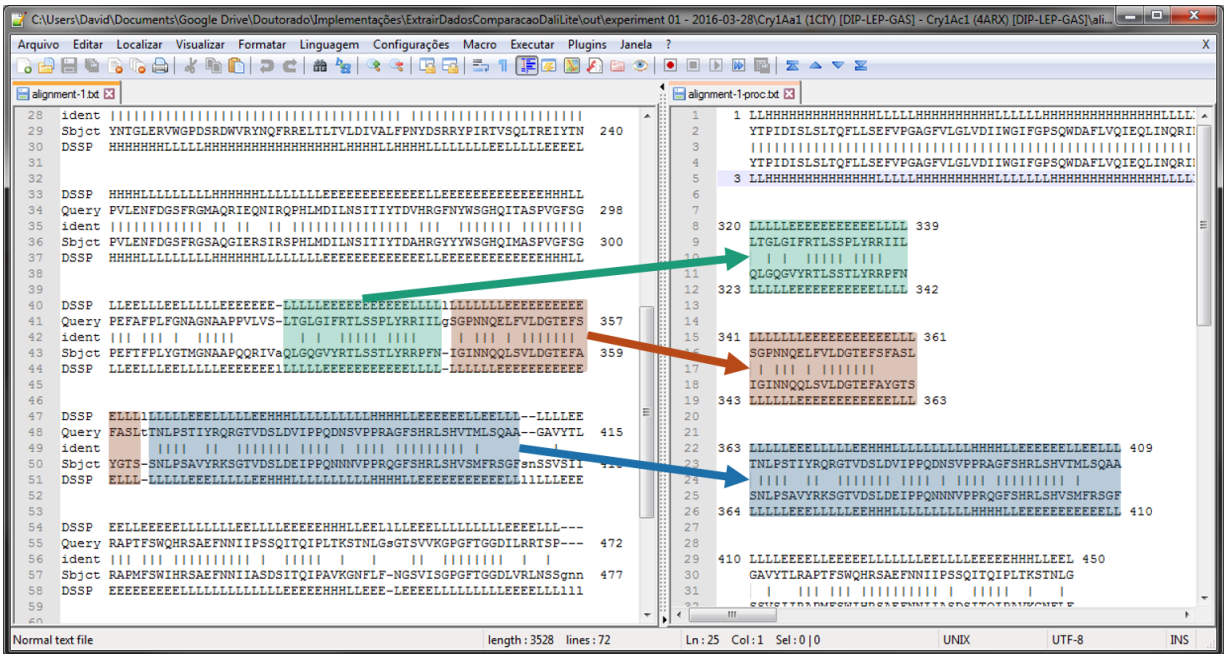
   Os **Recortes das Regiões com Interação com Receptores** são feitos com base nos dados fornecidos na Revisão da Literatura sobre a interação com receptores das proteínas Cry. Os recortes feitos correspondem à:

- Volta 2 do Domínio II da Proteína Cry1Aa1, intervalo [362; 372];
- Volta 3 do Domínio II da Proteína Cry1Aa1, intervalo [391; 400];
- Volta 8 do Domínio II da Proteína Cry1Aa1, intervalo [438; 447];
- Volta 2 do Domínio II da Proteína Cry1Ac1, intervalo [363; 373];
- Volta 3 do Domínio II da Proteína Cry1Ac1, intervalo [391; 400];
- *Loop* 1 do Domínio II da Proteína Cry3Aa1, intervalo [305; 321];
- Resíduos ^{157}R^{159} e ^{169}Y^{171} do Domínio I da Proteína Cry4Ba1, intervalo [158; 170];
- Resíduos ^{242}W^{244}, ^{245}F^{247}, ^{248}Y^{250} e ^{263}F^{265} do Domínio I da Proteína Cry4Ba1, intervalo [243; 264].

Com isso, os dados dos alinhamentos estão prontos para serem analisados;

7. **Dados para Análise:** O último passo do processo são os dados prontos para análise. Como exemplo, na Figura 88 pode ser visto como é o resultado do recorte de similaridade. Do lado esquerdo, pode-se ver o conteúdo do arquivo "alignment-1.txt", que contém o resultado do melhor alinhamento do Dali, enquanto do lado direito é apresentado o conteúdo do arquivo "alignment-1-proc.txt", que contém as seções com similaridade estrutural que foram recortadas. Há três setas, em verde, vermelho e azul, que mostram as correspondências entre os dois arquivos.

Figura 88 – Relação entre arquivo de resultados e arquivo de recorte de similaridades



Fonte: Elaborada pelo autor

Por fim, todos os resultados gerados anteriormente foram consolidados. Os arquivos dos resultados brutos podem ser encontrados no Apêndice A e os arquivos dos resultados finais podem ser encontrados no Apêndice B. As descrições de todas as implementações computacionais realizadas podem ser encontradas no Apêndice C.

5.9 ANÁLISE DE DADOS

A partir dos resultados finais gerados foi feita a análise dos mesmos. Os dados, bem como as imagens que os representam, serão apresentados no próximo Capítulo.

5.10 RESULTADOS E CONCLUSÕES

Os resultados e as conclusões desta pesquisa serão apresentados nos dois próximos Capítulos.

6 RESULTADOS E DISCUSSÃO

Os resultados desta pesquisa estão organizados em oito Tabelas (13, 14, 15, 16, 17, 18, 19, 20) e em oito Figuras (90, 91, 92, 93, 94, 95, 96, 97), que serão discutidas posteriormente. Cada Tabela faz par com a Figura seguinte, por exemplo, os dados da Tabela 13 são representados graficamente na Figura 90.

Cada par Tabela/Figura contém os dados de uma região ativa com receptores das proteínas Cry, dados esses baseados na Revisão da Literatura e indicados na legenda da Tabela e da Figura. Cada Tabela contém quatro colunas e oito conjuntos de dados. Na coluna "Referência" é indicada a seção da Figura que contém a representação gráfica do conjunto em questão, por exemplo, o primeiro conjunto de dados da Tabela 13 é representado graficamente na Figura 90a.

A segunda coluna de cada Tabela contém a indicação das duas proteínas que foram comparadas. No caso do primeiro conjunto de dados da Tabela 13, as proteínas comparadas foram a Cry1Aa1 e a Cry1Ac1. A coluna "Alinhamento" contém o resultado do alinhamento estrutural das proteínas indicadas para a região em questão. O alinhamento segue o seguinte padrão de representação:

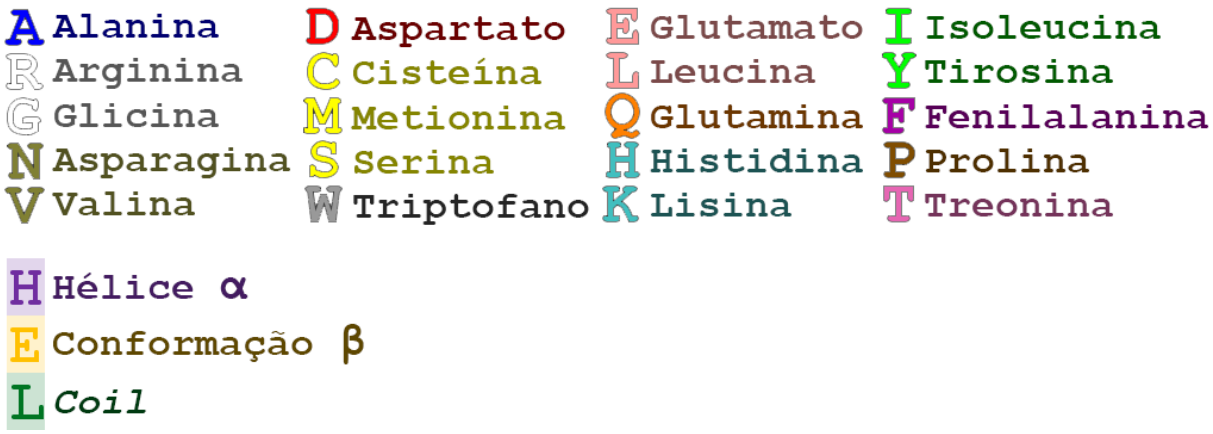
- Na primeira e na quinta linha são apresentados os componentes estruturais da região comparada, sendo que na primeira linha são os dados da primeira proteína e na quinta linha, os dados da segunda. Esses componentes são representados pelas letras H/h, em roxo, E/e, em amarelo, e L/l, em verde, sendo que denotam, respectivamente, hélices α, conformações β e *coils*;
- Na segunda e na terceira linha são apresentados os intervalos das estruturas primárias da primeira e da segunda proteína;
- Por fim, na linha três, a linha central, os aminoácidos iguais entre as duas proteínas, ou seja, as posições que possuem identidade, são indicados pelo caractere "|" (*pipe*).

Na quarta e última coluna, são apresentados os intervalos que representam as posições inicial e final dos intervalos de aminoácidos das duas proteínas da região comparada.

Em relação às Figuras, cada uma delas contém oito seções, indicadas pelas letras (a), (b), (c), (d), (e), (f), (g) e (h), sendo correspondentes aos dados da Tabela que as referenciam. Cada uma dessas seções é composta pela superposição estrutural da região comparada entre as duas proteínas, coloridas respectivamente em azul e vermelho, sendo que a proteína que possui os dados da região sensível à atividade com o receptor estará

sempre colorida em azul, bem como a separação das duas estruturas e sua respectiva representação linear, tanto em componentes estruturais, denotados com o mesmo padrão apresentado nas Tabelas, quanto nos aminoácidos que compõem aquela região, sendo que os identificadores de cada aminoácido estão coloridos de acordo com a Figura 89, que por sua vez reflete a coloração utilizada pelo *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996), usado neste trabalho para gerar as representações estruturais das proteínas.

Figura 89 – Legenda colorida usada para representar os resíduos de aminoácidos, baseada nas cores utilizadas pelo *software* VMD



Fonte: Elaborada pelo autor

Vale ressaltar que o VMD foi escolhido em detrimento de vários outros aplicativos de representação estrutural de proteínas[1], pois foi o que se mostrou, durante sua utilização, o mais acurado na geração de imagens representativas das estruturas das proteínas.

A seguir são apresentadas as Tabelas e Figuras explicadas anteriormente, bem como a discussão relativa aos dados representados por elas, sendo que essa discussão é baseada na inspeção visual realizada nos dados dos alinhamentos estruturais e nas representações gráficas das regiões de interesse, ou seja, regiões ativas com os receptores das proteínas Cry e que foram verificadas na Revisão da Literatura.

6.1 VOLTA 2 DO DOMÍNIO II DA PROTEÍNA Cry1Aa1

Os dados do alinhamento da Volta 2 do Domínio II da proteína Cry1Aa1, a qual é ativa contra as ordens de insetos *Diptera* e *Lepidoptera*, e contra a ordem *Gastropoda* do filo *Mollusca*, são apresentados na Tabela 13 e na Figura 90. Em relação à comparação com a proteína Cry1Ac1, ativa contra as mesmas ordens da proteína Cry1Aa1, é mostrado na Figura 90a que existem diferenças mínimas entre as duas, permitindo que se conclua

[1] Outros exemplos de *software* para representação estrutural de proteínas e/ou moléculas que podem ser citados são o PyMol (SCHRODINGER, 2015) e o Swiss-Pdb Viewer (GUEX; PEITSCH, 1997).

que essas pequenas modificações provavelmente não afetam suas especificidades, visto que essas proteínas são ativas contra as mesmas ordens.

O alinhamento com a proteína Cry2Aa1, ativa contra as ordens de insetos *Hemiptera* e *Lepidoptera*, é apresentado na Figura 90b. Verificando visualmente essa seção, pode-se notar que há diversas diferenças, tanto nos componentes estruturais, quanto na estrutura primária das duas proteínas comparadas. Essas diferenças podem implicar na especificidade da proteína Cry1Aa1 contra algumas espécies da ordem *Diptera*, visto que as duas proteínas, Cry1Aa1 e Cry2Aa1, são ativas contra a ordem *Lepidoptera*.

As comparações com as proteínas Cry3Aa1 e Cry3Bb1, ambas ativas contra a ordem *Coleoptera*, são apresentadas, respectivamente, na Figura 90c e na Figura 90d. Dado que a proteína Cry1Aa1 não compartilha atividade com a ordem afetada pelas proteínas Cry3Aa1 e Cry3Bb1, é possível afirmar que, provavelmente, as diferenças apresentadas nessas comparações mostram que essa região é importante na atividade das ordens atacadas pela toxina Cry1Aa1.

Em relação às comparações com as proteínas Cry4Aa1 e Cry4Ba1, ambas ativas contra a ordem *Diptera*, representadas respectivamente na Figura 90e e na Figura 90f, a comparação estrutural mostra que apesar de haver um compartilhamento de uma região de uma conformação β com a proteína Cry1Aa1, provavelmente essa região não deve ser responsável pela especificidade contra a ordem *Diptera*, visto que a proteína Cry1Aa1 também é tóxica contra essa ordem.

Na comparação com a proteína Cry5Ba1, ativa contra *Rhabditida*, que é uma ordem do filo *Nematoda* e está representada na Figura 90g, pode-se verificar que a região comparada é diferente entre as duas proteínas, o que é esperado, visto que ambas não compartilham atividade contra ordens iguais, ainda mais pela proteína Cry5Ba1 ser ativa contra nematóides ao invés de insetos. Como a proteína Cry5Ba1 é homóloga à proteína Cry5Aa1, a qual é ativa também contra a ordem de insetos *Hymenoptera*, provavelmente ela pode também ser ativa contra vespas/abelhas/formigas, entretanto a proteína Cry1Aa1 também não é ativa contra essa ordem.

A comparação com a proteína Cry8Ea1, ativa contra a ordem *Coleoptera* e representada na Figura 90h, mostra novamente diversas diferenças estruturais, o que provavelmente indica que essa região da proteína Cry1Aa1 deve ser importante na atividade contra *Diptera* e *Lepidoptera*, visto que ambas não compartilham atividade com ordens iguais.

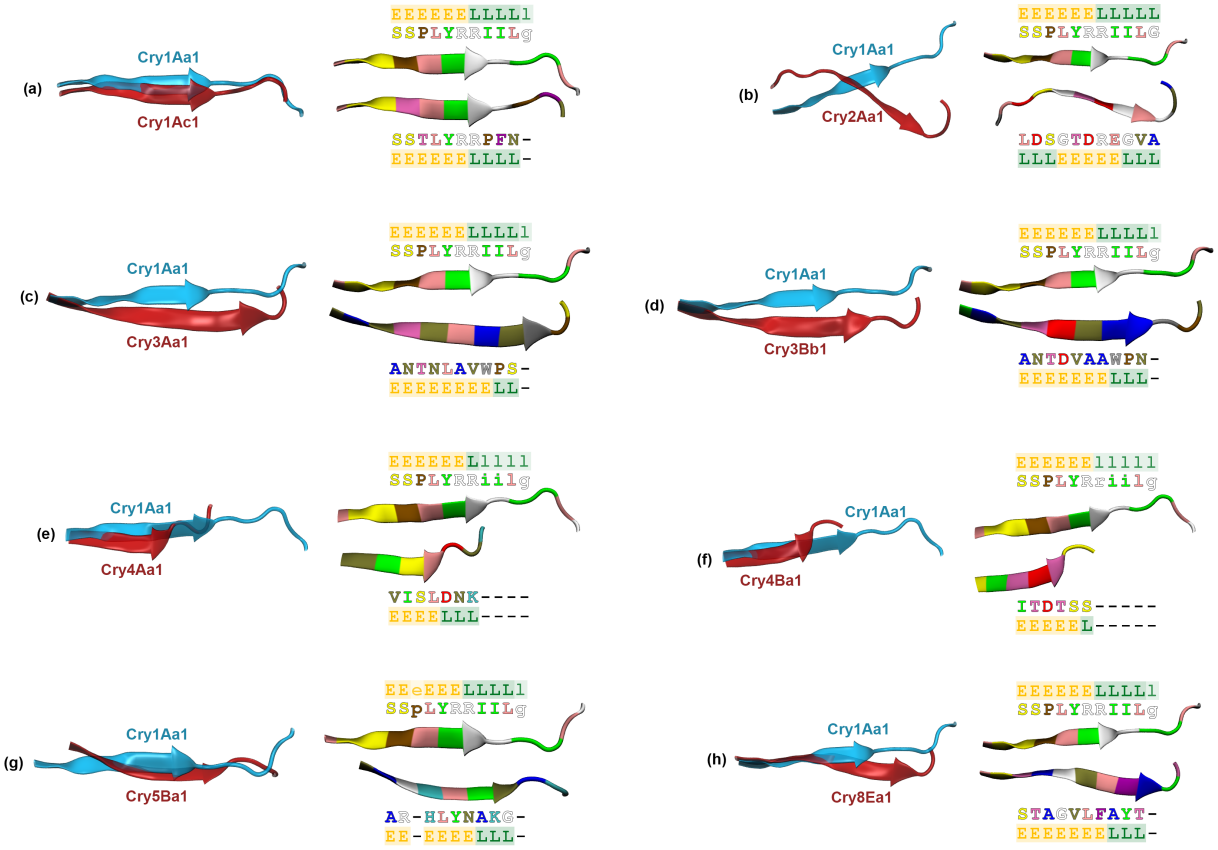
Por fim, vale ainda ressaltar que a proteína Cry1Aa1 é também ativa contra a ordem *Gastropoda*, entretanto, nenhuma das outras proteínas comparadas, com exceção da Cry1Ac1, possuem relação com essa ordem, então a mesma não foi e não será citada nas próximas comparações.

Tabela 13 – Alinhamentos da Volta 2 do Domínio II da proteína Cry1Aa1

Referência	**Proteínas**	**Alinhamento**	**Intervalos**
Figura 90a	Cry1Aa1	EEEEEELLLLl SSPLYRRIILg || ||||	[362; 372]
	Cry1Ac1	SSTLYRRPFN- EEEEEELLLL-	[363; 372]
Figura 90b	Cry1Aa1	EEEEEELLLLL SSPLYRRIILG |	[362; 372]
	Cry2Aa1	LDSGTDREGVA LLLEEEEELLL	[378; 388]
Figura 90c	Cry1Aa1	EEEEEELLLLl SSPLYRRIILg	[362; 372]
	Cry3Aa1	ANTNLAVWPS- EEEEEEEELL-	[404; 413]
Figura 90d	Cry1Aa1	EEEEEELLLLl SSPLYRRIILg	[362; 372]
	Cry3Bb1	ANTDVAAWPN- EEEEEEELLL-	[405; 414]
Figura 90e	Cry1Aa1	EEEEEELlllll SSPLYRRiilg |	[362; 372]
	Cry4Aa1	VISLDNK---- EEEELLL----	[426; 432]
Figura 90f	Cry1Aa1	EEEEEEllllll SSPLYRriilg	[362; 372]
	Cry4Ba1	ITDTSS----- EEEEEL-----	[383; 388]
Figura 90g	Cry1Aa1	EEeEEELLLLl SSpLYRRIILg	[362; 372]
	Cry5Ba1	AR-HLYNAKG- EE-EEEELLL-	[459; 467]
Figura 90h	Cry1Aa1	EEEEEELLLLl SSPLYRRIILg |	[362; 372]
	Cry8Ea1	STAGVLFAYT- EEEEEEELLL-	[409; 418]

Fonte: Elaborada pelo autor

Figura 90 – Representação gráfica dos alinhamentos da Volta 2 do Domínio II da proteína Cry1Aa1: (a) Alinhamento entre Cry1Aa1 e Cry1Ac1; (b) Alinhamento entre Cry1Aa1 e Cry2Aa1; (c) Alinhamento entre Cry1Aa1 e Cry3Aa1; (d) Alinhamento entre Cry1Aa1 e Cry3Bb1; (e) Alinhamento entre Cry1Aa1 e Cry4Aa1; (f) Alinhamento entre Cry1Aa1 e Cry4Ba1; (g) Alinhamento entre Cry1Aa1 e Cry5Ba1; e, (h) Alinhamento entre Cry1Aa1 e Cry8Ea1



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

6.2 VOLTA 3 DO DOMÍNIO II DA PROTEÍNA Cry1Aa1

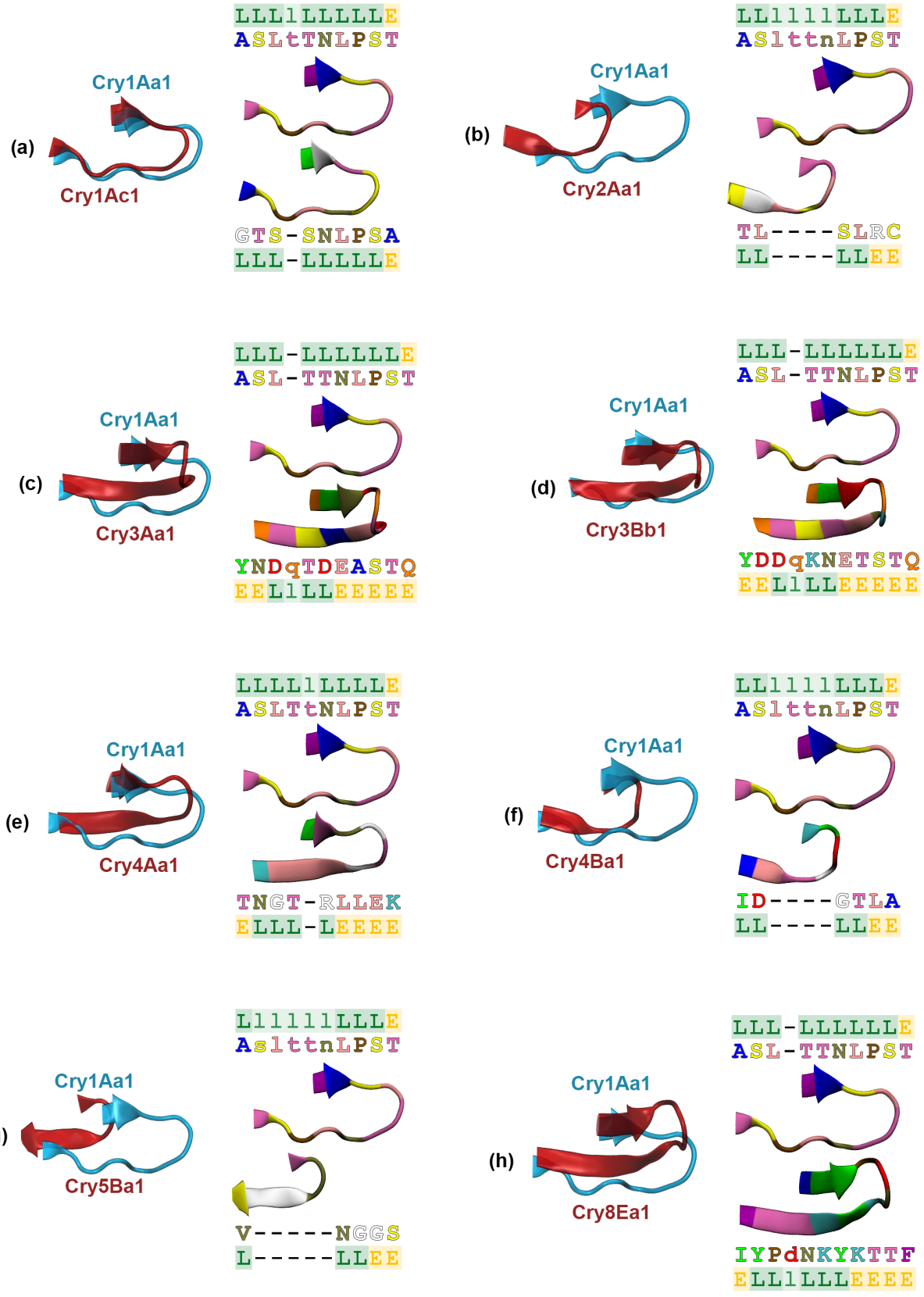
Os dados relativos à comparação da região da Volta 3 do Domínio II da proteína Cry1Aa1 com as outras proteínas são apresentados na Tabela 14 e na Figura 91. É possível verificar, pela inspeção desses dados, que com exceção da proteína Cry1Ac1, todas as outras proteínas diferem substancialmente nesta região em relação à proteína Cry1Aa1. Isso pode indicar que essa região é importante para a especificidade contra a ordem *Lepidoptera*, visto que essa é a ordem de insetos a qual as proteínas Cry1Aa1 e Cry1Ac1 são exclusivamente ativas em relação às proteínas comparadas.

Tabela 14 – Alinhamentos da Volta 3 do Domínio II da proteína Cry1Aa1

Referência	**Proteínas**	**Alinhamento**	**Intervalos**
Figura 91a	Cry1Aa1	LLL l LLLLLE ASLtTNLPST ||||	[391; 400]
	Cry1Ac1	GTS-SNLPSA LLL-LLLLLE	[391; 399]
Figura 91b	Cry1Aa1	LL llll LLLE ASlttnLPST	[391; 400]
	Cry2Aa1	TL----SLRC LL----LLEE	[401; 406]
Figura 91c	Cry1Aa1	LLL-LLLLLLE ASL-TTNLPST |	[391; 400]
	Cry3Aa1	YNDqTDEASTQ EEL l LLEEEEE	[427; 437]
Figura 91d	Cry1Aa1	LLL-LLLLLLE ASL-TTNLPST	[391; 400]
	Cry3Bb1	YDDqKNETSTQ EEL l LLEEEEE	[429; 439]
Figura 91e	Cry1Aa1	LLLL l LLLLE ASLTtNLPST | |	[391; 400]
	Cry4Aa1	TNGT-RLLEK ELLL-LEEEE	[448; 456]
Figura 91f	Cry1Aa1	LL llll LLLE ASlttnLPST	[391; 400]
	Cry4Ba1	ID----GTLA LL----LLEE	[401; 406]
Figura 91g	Cry1Aa1	L llllll LLLE AslttnLPST	[391; 400]
	Cry5Ba1	V-----NGGS L-----LLEE	[480; 484]
Figura 91h	Cry1Aa1	LLL-LLLLLLE ASL-TTNLPST	[391; 400]
	Cry8Ea1	IYPdNKYKTTF ELL l LLLEEEE	[432; 442]

Fonte: Elaborada pelo autor

Figura 91 – Representação gráfica dos alinhamentos da Volta 3 do Domínio II da proteína Cry1Aa1: (a) Alinhamento entre Cry1Aa1 e Cry1Ac1; (b) Alinhamento entre Cry1Aa1 e Cry2Aa1; (c) Alinhamento entre Cry1Aa1 e Cry3Aa1; (d) Alinhamento entre Cry1Aa1 e Cry3Bb1; (e) Alinhamento entre Cry1Aa1 e Cry4Aa1; (f) Alinhamento entre Cry1Aa1 e Cry4Ba1; (g) Alinhamento entre Cry1Aa1 e Cry5Ba1; e, (h) Alinhamento entre Cry1Aa1 e Cry8Ea1



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

6.3 VOLTA 8 DO DOMÍNIO II DA PROTEÍNA Cry1Aa1

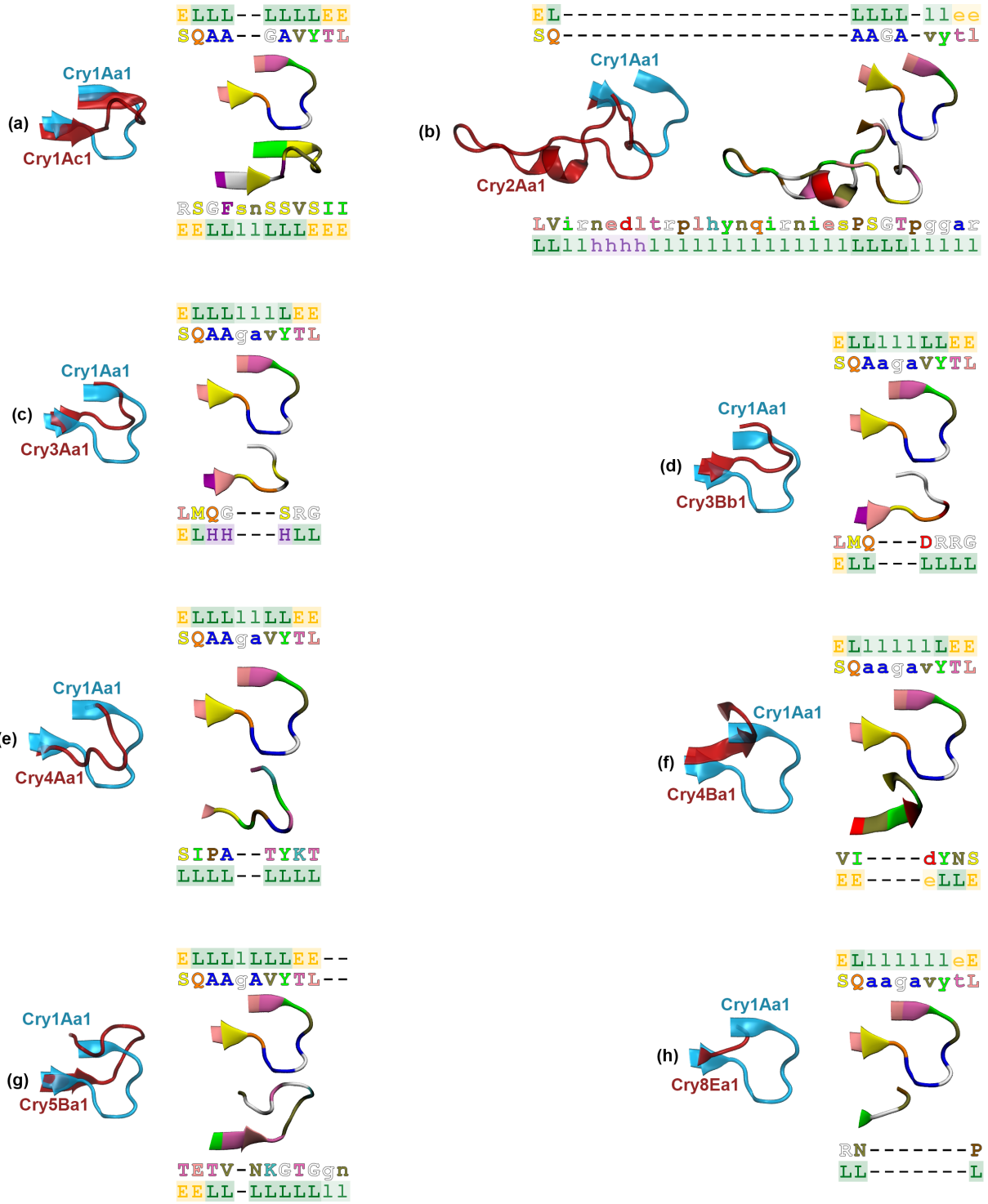
As diferenças estruturais entre as proteínas na região da Volta 8 do Domínio II da proteína Cry1Aa1, até mesmo com a proteína Cry1Ac1, que possui alta porcentagem de identidade com a Cry1Aa1, são evidentes nos dados apresentados na Tabela 15 e na Figura 92. Mesmo essa região sendo, de acordo com a Literatura, uma região importante na manifestação tóxica da proteína Cry1Aa1, provavelmente a mesma não deve ser essencial para que essa atividade ocorra em diversos insetos das ordens de insetos afetadas (*Diptera* e *Lepidoptera*), visto que a proteína Cry1Ac1 é ativa contra as mesmas ordens, permitindo concluir que, provavelmente, a proteína Cry1Aa1 seja mais específica a determinadas espécies das ordens afetadas do que a Cry1Ac1.

Tabela 15 – Alinhamentos da Volta 8 do Domínio II da proteína Cry1Aa1

Referência	Proteínas	Alinhamento	Intervalos
Figura 92a	Cry1Aa1	ELLL--LLLLEE SQAA--GAVYTL	[438; 447]
	Cry1Ac1	RSGFsnSSVSII EELLllLLLEEE	[437; 448]
Figura 92b	Cry1Aa1	EL----------LLLL-llee SQ----------AAGA-vytl	[438; 447]
	Cry2Aa1	LVirnedltrplhynqirniesPSGTpggar LLlhhhhlllllllllllllLLLLlllll	[431; 461]
Figura 92c	Cry1Aa1	ELLLlllLEE SQAAgavYTL	[438; 447]
	Cry3Aa1	LMQG---SRG ELHH---HLL	[480; 486]
Figura 92d	Cry1Aa1	ELLlllLLEE SQAagaVYTL	[438; 447]
	Cry3Bb1	LMQ---DRRG ELL---LLLL	[483; 489]
Figura 92e	Cry1Aa1	ELLLl1LLEE SQAAgaVYTL	[438; 447]
	Cry4Aa1	SIPA--TYKT LLLL--LLLL	[508; 515]
Figura 92f	Cry1Aa1	ELlllllLEE SQaagavYTL	[438; 447]
	Cry4Ba1	VI----dYNS EE----eLLE	[452; 457]
Figura 92g	Cry1Aa1	ELLLlLLLEE-- SQAAgAVYTL--	[438; 447]
	Cry5Ba1	TETV-NKGTGgn EELL-LLLLL11	[519; 529]
Figura 92h	Cry1Aa1	ELlllllleE SQaagavytL	[438; 447]
	Cry8Ea1	RN-------P LL-------L	[486; 488]

Fonte: Elaborada pelo autor

Figura 92 – Representação gráfica dos alinhamentos da Volta 8 do Domínio II da proteína Cry1Aa1: (a) Alinhamento entre Cry1Aa1 e Cry1Ac1; (b) Alinhamento entre Cry1Aa1 e Cry2Aa1; (c) Alinhamento entre Cry1Aa1 e Cry3Aa1; (d) Alinhamento entre Cry1Aa1 e Cry3Bb1; (e) Alinhamento entre Cry1Aa1 e Cry4Aa1; (f) Alinhamento entre Cry1Aa1 e Cry4Ba1; (g) Alinhamento entre Cry1Aa1 e Cry5Ba1; e, (h) Alinhamento entre Cry1Aa1 e Cry8Ea1



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

6.4 VOLTAS 2 E 3 DO DOMÍNIO II DA PROTEÍNA Cry1Ac1

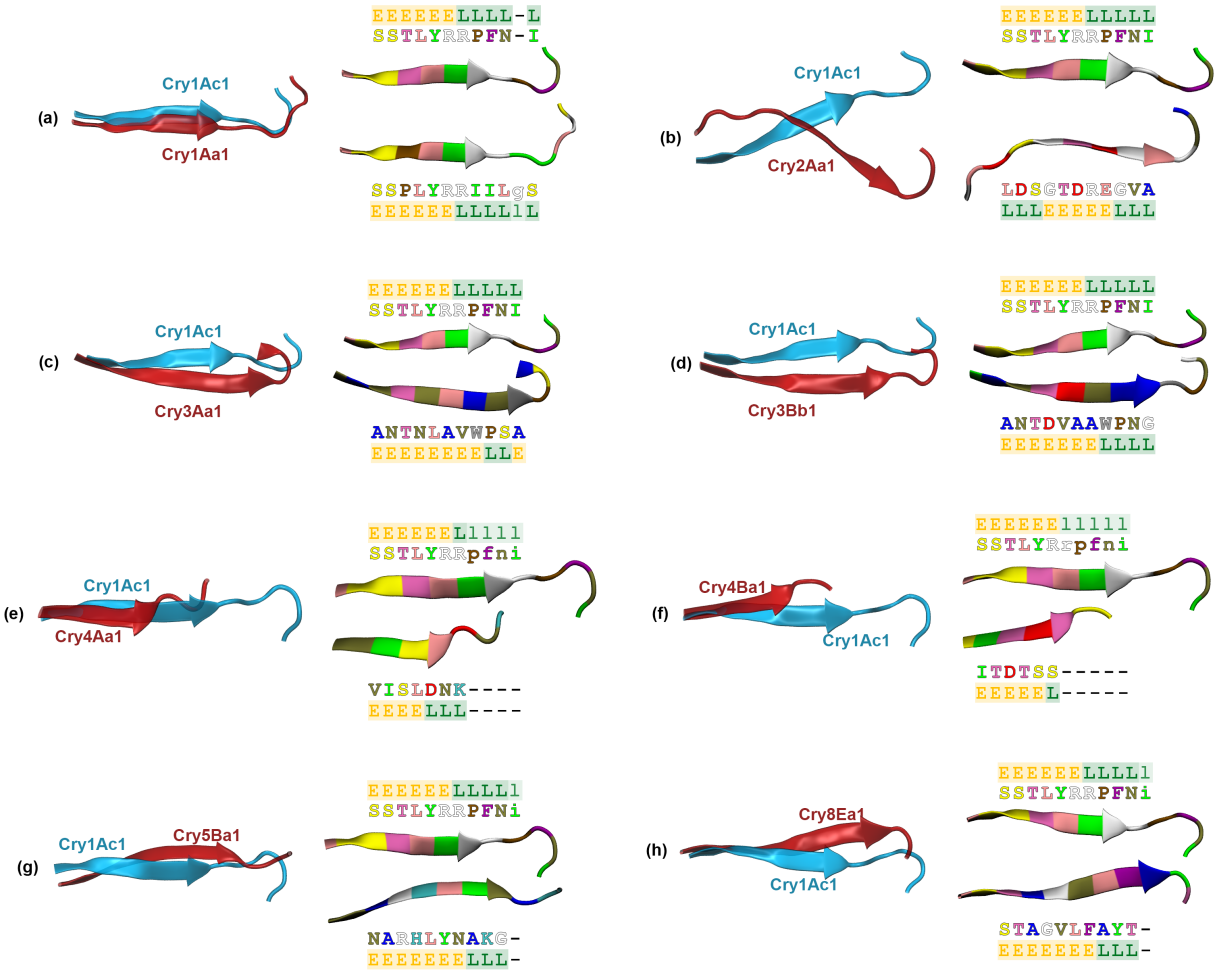
Os resultados apresentados nos pares Tabela 16/Figura 93 e Tabela 17/Figura 94, relativos respectivamente às Voltas 2 e 3 do Domínio II da proteína Cry1Ac1, ao serem verificados, pôde-se constatar a correlação existente com as regiões das Voltas 2 e 3 do Domínio II da proteína Cry1Aa1, visto que as proteínas Cry1Ac1 e Cry1Aa1 têm alta porcentagem de identidade. Sendo assim, as mesmas observações feitas e implicações levantadas nas Seções anteriores se aplicam para essas duas regiões da proteína Cry1Ac1.

Tabela 16 – Alinhamentos da Volta 2 do Domínio II da proteína Cry1Ac1

Referência	Proteínas	Alinhamento	Intervalos
Figura 93a	Cry1Ac1	EEEEEELLLL-L SSTLYRRPFN-I || |||| SSPLYRRIILgS EEEEEELLLL1L	[363; 373] [362; 373]
Figura 93b	Cry1Ac1	EEEEEELLLLL SSTLYRRPFNI | LDSGTDREGVA LLLEEEEELLL	[363; 373] [378; 388]
Figura 93c	Cry1Ac1	EEEEEELLLLL SSTLYRRPFNI | ANTNLAVWPSA EEEEEEEELLE	[363; 373] [404; 414]
Figura 93d	Cry1Ac1	EEEEEELLLLL SSTLYRRPFNI | | ANTDVAAWPNG EEEEEEELLLL	[363; 373] [405; 415]
Figura 93e	Cry1Ac1	EEEEEEL1111 SSTLYRRpfni | VISLDNK---- EEEELLL----	[363; 373] [426; 432]
Figura 93f	Cry1Ac1	EEEEEE11111 SSTLYRrpfni ITDTSS----- EEEEEL-----	[363; 373] [383; 388]
Figura 93g	Cry1Ac1	EEEEEELLLL1 SSTLYRRPFNi NARHLYNAKG- EEEEEEELLL-	[363; 373] [458; 467]
Figura 93h	Cry1Ac1	EEEEEELLLL1 SSTLYRRPFNi | STAGVLFAYT- EEEEEEELLL-	[363; 373] [409; 418]

Fonte: Elaborada pelo autor

Figura 93 – Representação gráfica dos alinhamentos da Volta 2 do Domínio II da proteína Cry1Ac1: (a) Alinhamento entre Cry1Ac1 e Cry1Aa1; (b) Alinhamento entre Cry1Ac1 e Cry2Aa1; (c) Alinhamento entre Cry1Ac1 e Cry3Aa1; (d) Alinhamento entre Cry1Ac1 e Cry3Bb1; (e) Alinhamento entre Cry1Ac1 e Cry4Aa1; (f) Alinhamento entre Cry1Ac1 e Cry4Ba1; (g) Alinhamento entre Cry1Ac1 e Cry5Ba1; e, (h) Alinhamento entre Cry1Ac1 e Cry8Ea1



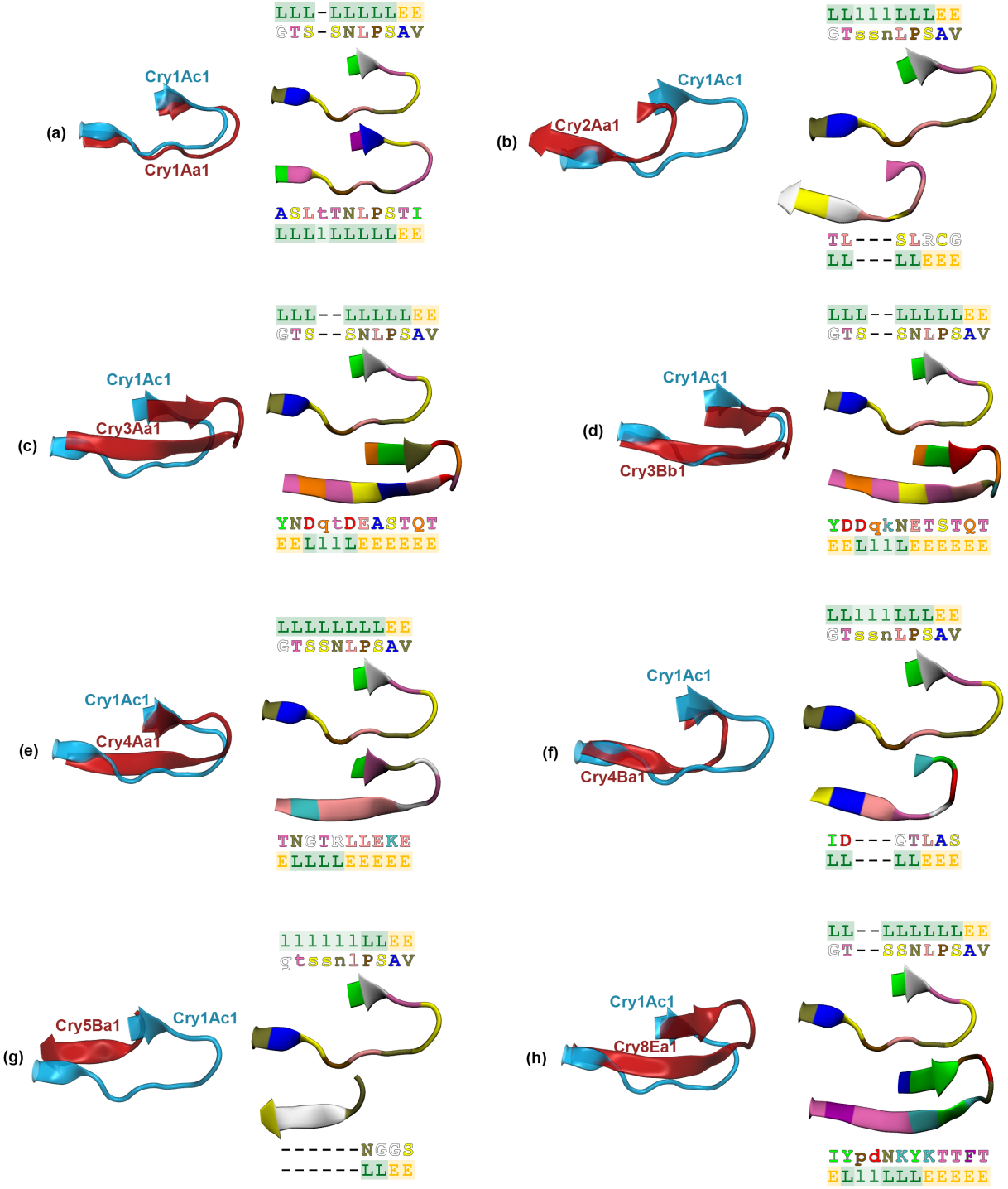
Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

Tabela 17 – Alinhamentos da Volta 3 do Domínio II da proteína Cry1Ac1

Referência	**Proteínas**	**Alinhamento**	**Intervalos**
Figura 94a	Cry1Ac1	LLL-LLLLLEE GTS-SNLPSAV	[391; 400]
	Cry1Aa1	ASLtTNLPSTI LLLlLLLLLEE	[391; 401]
Figura 94b	Cry1Ac1	LLlllLLLEE GTssnLPSAV	[391; 400]
	Cry2Aa1	TL---SLRCG LL---LLEEE	[401; 407]
Figura 94c	Cry1Ac1	LLL--LLLLLEE GTS--SNLPSAV	[391; 400]
	Cry3Aa1	YNDqtDEASTQT EELllLEEEEEE	[427; 438]
Figura 94d	Cry1Ac1	LLL--LLLLLEE GTS--SNLPSAV	[391; 400]
	Cry3Bb1	YDDqkNETSTQT EELllLEEEEEE	[429; 440]
Figura 94e	Cry1Ac1	LLLLLLLLEE GTSSNLPSAV	[391; 400]
	Cry4Aa1	TNGTRLLEKE ELLLLEEEEE	[448; 457]
Figura 94f	Cry1Ac1	LLlllLLLEE GTssnLPSAV	[391; 400]
	Cry4Ba1	ID---GTLAS LL---LLEEE	[401; 407]
Figura 94g	Cry1Ac1	llllllLLEE gtssnlPSAV	[391; 400]
	Cry5Ba1	------NGGS ------LLEE	[481; 484]
Figura 94h	Cry1Ac1	LL--LLLLLLEE GT--SSNLPSAV	[391; 400]
	Cry8Ea1	IYpdNKYKTTFT ELllLLLEEEEE	[432; 443]

Fonte: Elaborada pelo autor

Figura 94 – Representação gráfica dos alinhamentos da Volta 3 do Domínio II da proteína Cry1Ac1: (a) Alinhamento entre Cry1Ac1 e Cry1Aa1; (b) Alinhamento entre Cry1Ac1 e Cry2Aa1; (c) Alinhamento entre Cry1Ac1 e Cry3Aa1; (d) Alinhamento entre Cry1Ac1 e Cry3Bb1; (e) Alinhamento entre Cry1Ac1 e Cry4Aa1; (f) Alinhamento entre Cry1Ac1 e Cry4Ba1; (g) Alinhamento entre Cry1Ac1 e Cry5Ba1; e, (h) Alinhamento entre Cry1Ac1 e Cry8Ea1



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

6.5 *LOOP* 1 DO DOMÍNIO II DA PROTEÍNA Cry3Aa1

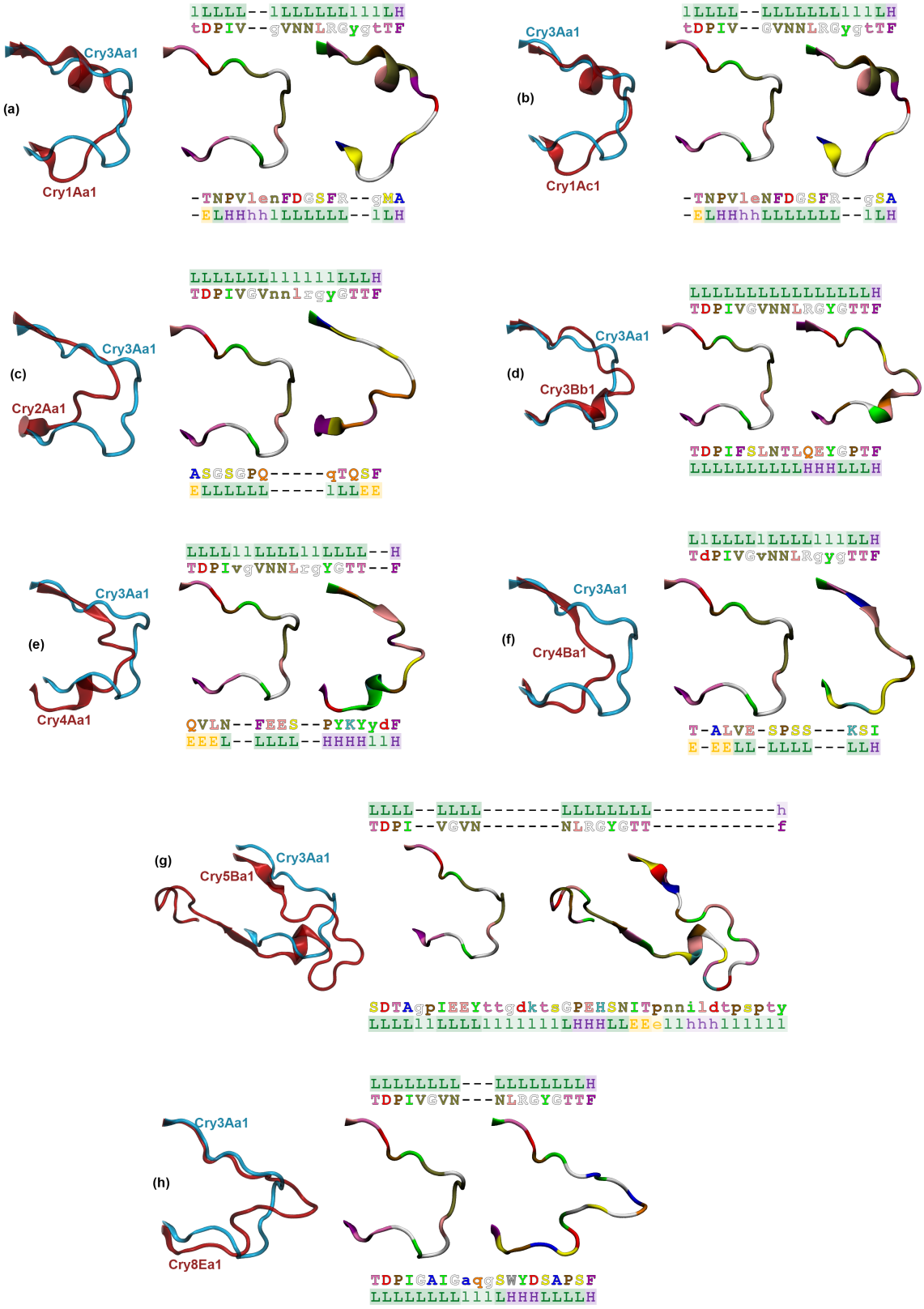
Os dados relacionados à comparação estrutural da região do *Loop* 1 do Domínio II da proteína Cry3Aa1 são apresentados na Tabela 18 e na Figura 95. Entre todas as proteínas comparadas, as proteínas Cry3Aa1, Cry3Bb1 e Cry8Ea1 compartilham atividade contra a ordem de insetos *Coleoptera*. Essa região parece ser importante para essa manifestação tóxica, visto que essas três proteínas são mais similares nessa região em relação às outras. Essa verificação pode ser vista nas seções (d) e (h) da Figura 95. Pode-se notar que em comparação com a proteína Cry3Bb1, representada no quarto conjunto de dados da Tabela 18 e Figura 95d, existem dez identidades entre as duas proteínas, além de existir similaridade estrutural em ambas as extremidades. Em relação à proteína Cry8Ea1, Figura 95h, há identidade nos quatro primeiros resíduos dessa região, além de estruturalmente serem parecidas até o início da inserção dos *gaps* para o alinhamento estrutural. Ao se analisar essa relação, pode-se afirmar que, provavelmente, os quatro primeiros aminoácidos, `TDPI`, têm participação importante na manifestação tóxica dessas proteínas em relação à ordem *Coleoptera*, visto que as proteínas Cry3Aa1, Cry3Bb1 e Cry8Ea1 compartilham identidade nesses quatro aminoácidos, além de terem conformação praticamente idêntica nessa região.

Tabela 18 – Alinhamentos do *Loop* 1 do Domínio II da proteína Cry3Aa1

Referência	**Proteínas**	**Alinhamento**	**Intervalos**
Figura 95a	Cry3Aa1	lLLLL--lLLLLLLlllLH tDPIV--gVNNLRGygtTF	[305; 321]
	Cry1Aa1	-TNPVlenFDGSFR--gMA -ELHHhhlLLLLLL--lLH	[269; 284]
Figura 95b	Cry3Aa1	lLLLL--LLLLLLLlllLH tDPIV--GVNNLRGygtTF	[305; 321]
	Cry1Ac1	-TNPVleNFDGSFR--gSA -ELHHhhLLLLLLL--lLH	[269; 284]
Figura 95c	Cry3Aa1	LLLLLLLllllllLLLH TDPIVGVnnlrgyGTTF	[305; 321]
	Cry2Aa1	ASGSGPQ-----qTQSF ELLLLLL-----lLLEE	[277; 288]
Figura 95d	Cry3Aa1	LLLLLLLLLLLLLLLLH TDPIVGVNNLRGYGTTF	[305; 321]
	Cry3Bb1	TDPIFSLNTLQEYGPTF LLLLLLLLLLHHHLLLH	[306; 322]
Figura 95e	Cry3Aa1	LLLLllLLLLllLLLL--H TDPIvgVNNLrgYGTT--F	[305; 321]
	Cry4Aa1	QVLN--FEES--PYKYydF EEEL--LLLL--HHHHllH	[332; 346]
Figura 95f	Cry3Aa1	LlLLLLlLLLLlllLLH TdPIVGvNNLRgygTTF	[305; 321]
	Cry4Ba1	T-ALVE-SPSS---KSI E-EELL-LLLL---LLH	[293; 304]
Figura 95g	Cry3Aa1	LLLL--LLLL-------LLLLLLLL----------h TDPI--VGVN-------NLRGYGTT----------f	[305; 321]
	Cry5Ba1	SDTAgpIEEYttgdktsGPEHSNITpnnildtpspty LLLLllLLLLlllllllLHHHLLEEellhhhllllll	[345; 381]
Figura 95h	Cry3Aa1	LLLLLLLL---LLLLLLLLH TDPIVGVN---NLRGYGTTF	[305; 321]
	Cry8Ea1	TDPIGAIGaqgSWYDSAPSF LLLLLLLLlllLHHHLLLLH	[307; 326]

Fonte: Elaborada pelo autor

Figura 95 – Representação gráfica dos alinhamentos do *Loop* 1 do Domínio II da proteína Cry3Aa1: (a) Alinhamento entre Cry3Aa1 e Cry1Aa1; (b) Alinhamento entre Cry3Aa1 e Cry1Ac1; (c) Alinhamento entre Cry3Aa1 e Cry2Aa1; (d) Alinhamento entre Cry3Aa1 e Cry3Bb1; (e) Alinhamento entre Cry3Aa1 e Cry4Aa1; (f) Alinhamento entre Cry3Aa1 e Cry4Ba1; (g) Alinhamento entre Cry3Aa1 e Cry5Ba1; e, (h) Alinhamento entre Cry3Aa1 e Cry8Ea1



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

6.6 RESÍDUOS ^{157}R^{159} E ^{169}Y^{171} DO DOMÍNIO I DA PROTEÍNA Cry4Ba1

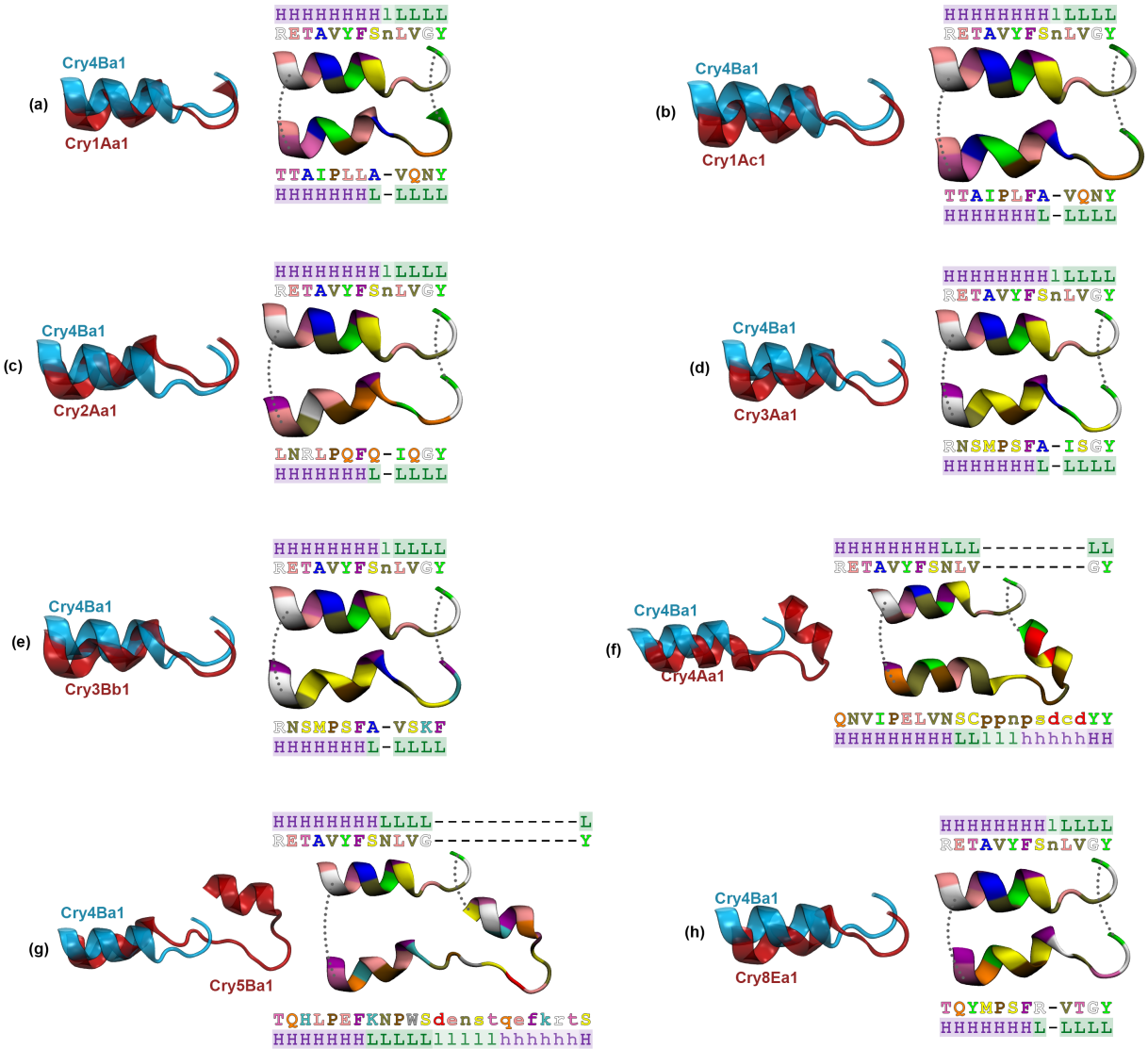
Experimentalmente, de acordo com a Literatura, foi verificado que existem alguns aminoácidos que são importantes para a atividade tóxica da proteína Cry4Ba1, sendo que os mesmos estão situados no Domínio I, responsável pela inserção da proteína, em sua conformação tóxica, na membrana intestinal e na formação dos poros nos insetos. Dois desses resíduos são ^{157}R^{159} e ^{169}Y^{171}. Nos dados relativos à comparação desses resíduos, apresentados na Tabela 19 e na Figura 96, foram recortadas as regiões que compreendem esses aminoácidos e todos os aminoácidos que estão entre os dois. Foi escolhida essa abordagem para permitir que essa região fosse melhor observada estrutural e graficamente. Como pode ser visto na Figura 96, todas as proteínas, com exceção da Cry4Aa1 (Figura 96f) e da Cry5Ba1 (Figura 96g) compartilham, nessa região, a mesma estrutura secundária, isto é, uma hélice α seguida de um *coil*. O aminoácido ^{169}Y^{171} parece ser mais importante que o primeiro, visto que o mesmo é mais conservado nas proteínas comparadas. Nos casos onde há modificação, ou seja, na proteína Cry3Bb1 com uma Fenilalanina e na Cry5Ba1 com uma Serina, pode-se verificar que esses dois aminoácidos têm propriedades químicas similares à Tirosina da proteína Cry4Ba1. Os aminoácidos ^{157}R^{159} e ^{169}Y^{171} provavelmente têm papel mais específico em relação a estabilidade e atividade das proteínas Cry comparadas no que tange ao seu modo de ação geral e não especificamente à especificidade, visto que essa região é similar estruturalmente em quase todas as proteínas comparadas.

Tabela 19 – Alinhamento dos Resíduos R e Y do Domínio I da proteína Cry4Ba1

Referência	Proteínas	Alinhamento	Intervalos
Figura 96a	Cry4Ba1	HHHHHHHHlLLLL RETAVYFSnLVGY |	[158; 170]
	Cry1Aa1	TTAIPLLA-VQNY HHHHHHHL-LLLL	[142; 153]
Figura 96b	Cry4Ba1	HHHHHHHHlLLLL RETAVYFSnLVGY |	[158; 170]
	Cry1Ac1	TTAIPLFA-VQNY HHHHHHHL-LLLL	[142; 153]
Figura 96c	Cry4Ba1	HHHHHHHHlLLLL RETAVYFSnLVGY | ||	[158; 170]
	Cry2Aa1	LNRLPQFQ-IQGY HHHHHHHL-LLLL	[158; 169]
Figura 96d	Cry4Ba1	HHHHHHHHlLLLL RETAVYFSnLVGY |	||	[158; 170]
	Cry3Aa1	RNSMPSFA-ISGY HHHHHHHL-LLLL	[178; 189]
Figura 96e	Cry4Ba1	HHHHHHHHlLLLL RETAVYFSnLVGY |	[158; 170]
	Cry3Bb1	RNSMPSFA-VSKF HHHHHHHL-LLLL	[179; 190]
Figura 96f	Cry4Ba1	HHHHHHHHLLL--------LL RETAVYFSNLV--------GY |	[158; 170]
	Cry4Aa1	QNVIPELVNSCppnpsdcdYY HHHHHHHHHLLlllhhhhhHH	[182; 202]
Figura 96g	Cry4Ba1	HHHHHHHHLLLL----------L RETAVYFSNLVG----------Y |	[158; 170]
	Cry5Ba1	TQHLPEFKNPWSdenstqefkrtS HHHHHHHLLLLLlllllhhhhhhH	[205; 228]
Figura 96h	Cry4Ba1	HHHHHHHHlLLLL RETAVYFSnLVGY | ||	[158; 170]
	Cry8Ea1	TQYMPSFR-VTGY HHHHHHHL-LLLL	[180; 191]

Fonte: Elaborada pelo autor

Figura 96 – Representação gráfica dos alinhamentos dos Resíduos $^{157}R^{159}$ e $^{169}Y^{171}$ do Domínio I da proteína Cry4Ba1: (a) Alinhamento entre Cry4Ba1 e Cry1Aa1; (b) Alinhamento entre Cry4Ba1 e Cry1Ac1; (c) Alinhamento entre Cry4Ba1 e Cry2Aa1; (d) Alinhamento entre Cry4Ba1 e Cry3Aa1; (e) Alinhamento entre Cry4Ba1 e Cry3Bb1; (f) Alinhamento entre Cry4Ba1 e Cry4Aa1; (g) Alinhamento entre Cry4Ba1 e Cry5Ba1; e, (h) Alinhamento entre Cry4Ba1 e Cry8Ea1



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

6.7 RESÍDUOS $^{242}W^{244}$, $^{245}F^{247}$, $^{248}Y^{250}$ E $^{263}F^{265}$ DO DOMÍNIO I DA PROTEÍNA Cry4Ba1

A abordagem representativa dos resíduos $^{242}W^{244}$, $^{245}F^{247}$, $^{248}Y^{250}$ e $^{263}F^{265}$ do Domínio I da proteína Cry4Ba1 é a mesma utilizada na seção anterior, sendo que os dados das comparações podem ser verificados na Tabela 20 e na Figura 97. A região compreendida

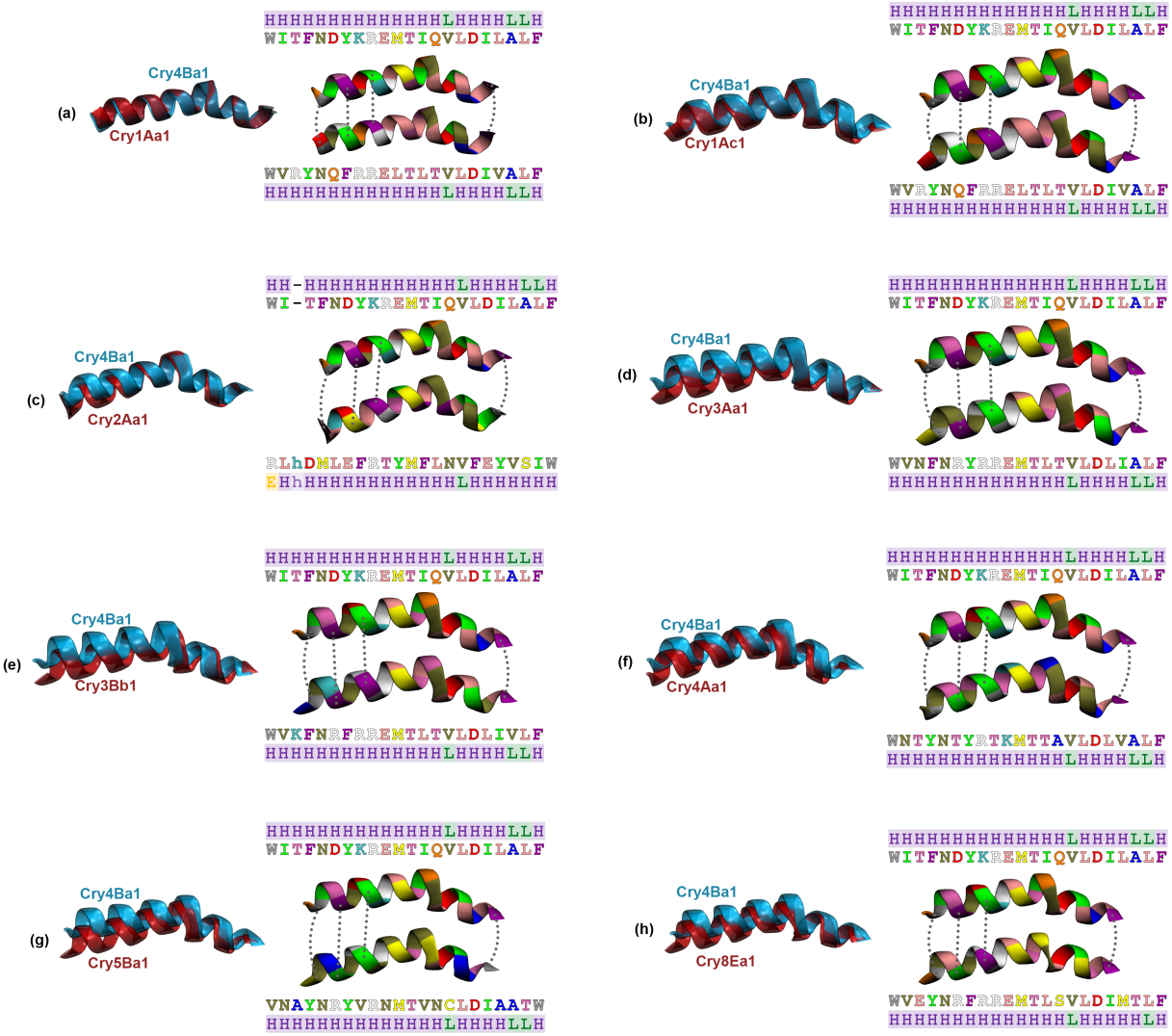
entre esses resíduos, inserida no final do Domínio I das proteínas, compartilha diversas identidades entre as proteínas comparadas, tendo provavelmente a mesma implicação que a região apresentada anteriormente, ou seja, importância na estabilidade estrutural das proteínas e no modo de ação geral do Domínio I das proteínas Cry de três domínios, visto a conservação existente nessa região. Pode-se notar que no final dessa região existem quatro resíduos, `VLDI`, que são conservados quase que totalmente em todas as proteínas comparadas, mostrando que essa região é importante para todas as proteínas.

Tabela 20 – Alinhamento dos Resíduos W, F, Y e F do Domínio I da proteína Cry4Ba1

Referência	Proteínas	Alinhamento	Intervalos
Figura 97a	Cry4Ba1	HHHHHHHHHHHHHHLHHHHLLH WITFNDYKREMTIQVLDILALF | | || | |||| |||	[243; 264]
	Cry1Aa1	WVRYNQFRRELTLTVLDIVALF HHHHHHHHHHHHHHLHHHHLLH	[226; 247]
Figura 97b	Cry4Ba1	HHHHHHHHHHHHHHLHHHHLLH WITFNDYKREMTIQVLDILALF | | || | |||| |||	[243; 264]
	Cry1Ac1	WVRYNQFRRELTLTVLDIVALF HHHHHHHHHHHHHHLHHHHLLH	[226; 247]
Figura 97c	Cry4Ba1	HH-HHHHHHHHHHHLHHHHLLH WI-TFNDYKREMTIQVLDILALF | |	[243; 264]
	Cry2Aa1	RLhDMLEFRTYMFLNVFEYVSIW EHhHHHHHHHHHHHLHHHHHHH	[237; 259]
Figura 97d	Cry4Ba1	HHHHHHHHHHHHHHLHHHHLLH WITFNDYKREMTIQVLDILALF | || | |||| ||| |||	[243; 264]
	Cry3Aa1	WVNFNRYRREMTLTVLDLIALF HHHHHHHHHHHHHHLHHHHLLH	[262; 283]
Figura 97e	Cry4Ba1	HHHHHHHHHHHHHHLHHHHLLH WITFNDYKREMTIQVLDILALF | || |||| ||| ||	[243; 264]
	Cry3Bb1	WVKFNRFRREMTLTVLDLIVLF HHHHHHHHHHHHHHLHHHHLLH	[263; 284]
Figura 97f	Cry4Ba1	HHHHHHHHHHHHHHLHHHHLLH WITFNDYKREMTIQVLDILALF | | | | || ||| |||	[243; 264]
	Cry4Aa1	WNTYNTYRTKMTTAVLDLVALF HHHHHHHHHHHHHHLHHHHLLH	[289; 310]
Figura 97g	Cry4Ba1	HHHHHHHHHHHHHHLHHHHLLH WITFNDYKREMTIQVLDILALF | | | || ||| |	[243; 264]
	Cry5Ba1	VNAYNRYVRNMTVNCLDIAATW HHHHHHHHHHHHHHLHHHHLLH	[302; 323]
Figura 97h	Cry4Ba1	HHHHHHHHHHHHHHLHHHHLLH WITFNDYKREMTIQVLDILALF | | |||| |||| ||	[243; 264]
	Cry8Ea1	WVEYNRFRREMTLSVLDIMTLF HHHHHHHHHHHHHHLHHHHHLH	[264; 285]

Fonte: Elaborada pelo autor

Figura 97 – Representação gráfica dos alinhamentos dos Resíduos ^{242}W^{244}, ^{245}F^{247}, ^{248}Y^{250} e ^{263}F^{265} do Domínio I da proteína Cry4Ba1: (a) Alinhamento entre Cry4Ba1 e Cry1Aa1; (b) Alinhamento entre Cry4Ba1 e Cry1Ac1; (c) Alinhamento entre Cry4Ba1 e Cry2Aa1; (d) Alinhamento entre Cry4Ba1 e Cry3Aa1; (e) Alinhamento entre Cry4Ba1 e Cry3Bb1; (f) Alinhamento entre Cry4Ba1 e Cry4Aa1; (g) Alinhamento entre Cry4Ba1 e Cry5Ba1; e, (h) Alinhamento entre Cry4Ba1 e Cry8Ea1



Fonte: Elaborada pelo autor utilizando o *software* VMD (HUMPHREY; DALKE; SCHULTEN, 1996) versão 1.9.3

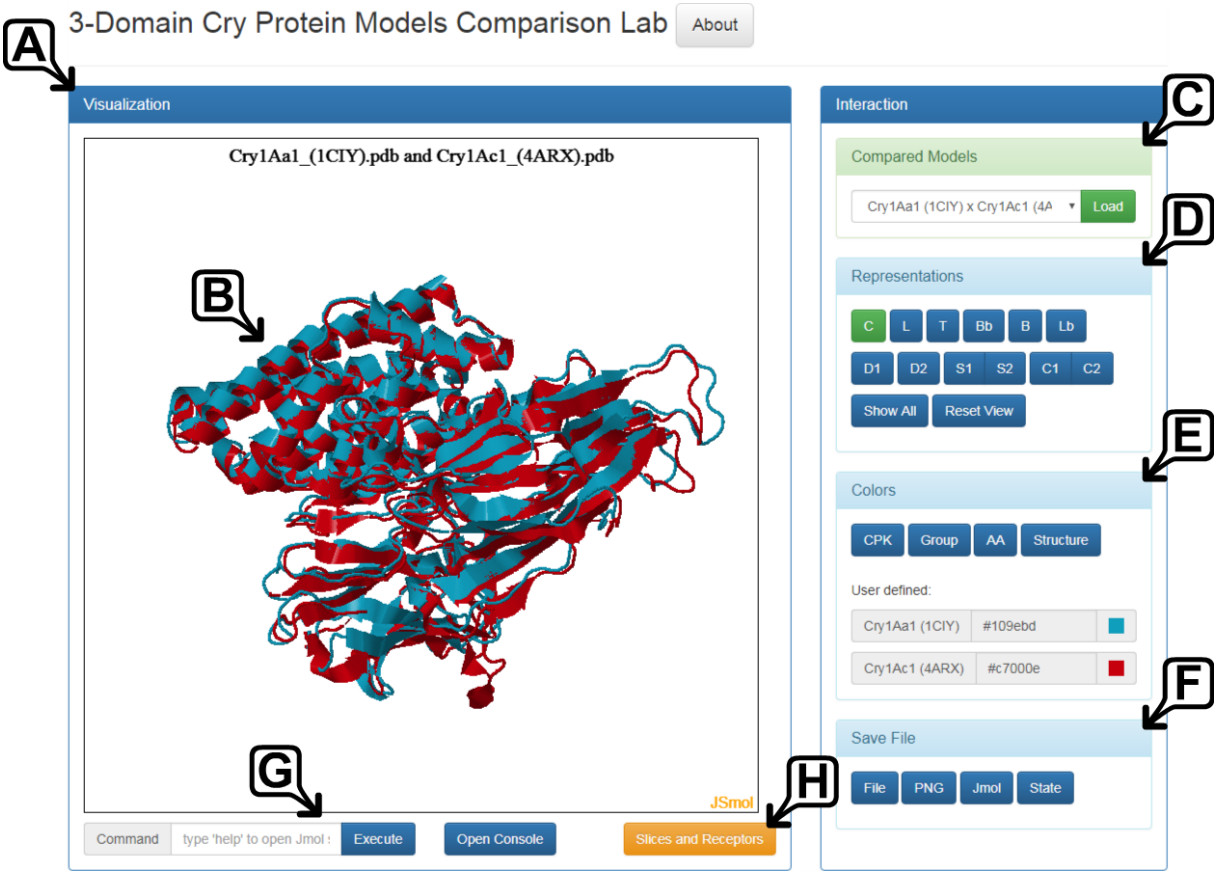
Os resultados apresentados e discutidos podem ser verificados em tempo real por meio da utilização de uma aplicação *Web* que foi desenvolvida. Essa aplicação será apresentada sucintamente na próxima Seção.

6.8 FERRAMENTA DE VISUALIZAÇÃO DOS RESULTADOS *3-DOMAIN CRY PROTEIN MODELS COMPARISON LAB*

Com o objetivo de facilitar e permitir a visualização dos dados das comparações estruturais entre as nove proteínas Cry processadas, foi criada uma aplicação *Web*, ou seja, um programa de computador acessível via navegador de Internet. Essa aplicação foi intitulada "*3-Domain Cry Protein Models Comparison Lab*" e pode ser acessada pelo endereço <https://sbv.ifsp.edu.br/CryProteinModelsComparisonLab/>.

Essa ferramenta, como já informado, pode ser utilizada para visualizar os dados do experimento que foi conduzido nesse trabalho. Na Figura 98 é apresentada a interface gráfica principal da ferramenta. A seta "A" aponta para o quadro de visualização, onde os detalhes dos alinhamentos entre as proteínas podem ser visualizados. No exemplo, é apresentada a superposição estrutural das proteínas Cry1Aa1 e Cry1Ac1, indicada pela seta "B". Para escolher qual alinhamento estrutural deve ser exibido, utiliza-se a caixa de seleção apontada pela seta "C" e então clica-se no botão "*Load*". O painel de representações ("*Representations*"), indicado pela seta "D", é utilizado para modificar a representação gráfica usada para desenhar as proteínas. Ainda com o objetivo de alterar a representação gráfica das moléculas, pode-se utilizar o painel "*Colors*", indicado pela seta "E", que permite modificar a cor aplicada na visualização das estruturas. O painel "*Save File*", apontado pela seta "F", permite que o usuário salve os dados da visualização em alguns formatos. A seta "G" aponta para a caixa de texto que permite aos usuários inserirem comandos, de acordo com a linguagem de *script* interativa Jmol/JSmol, para terem um maior controle de como as estruturas são exibidas. Por fim, a seta "H" aponta para o botão "*Slices and Receptors*" que, ao ser clicado, exibe todos os dados obtidos a partir da comparação estrutural selecionada e que foi previamente calculada pelo algoritmo Dali.

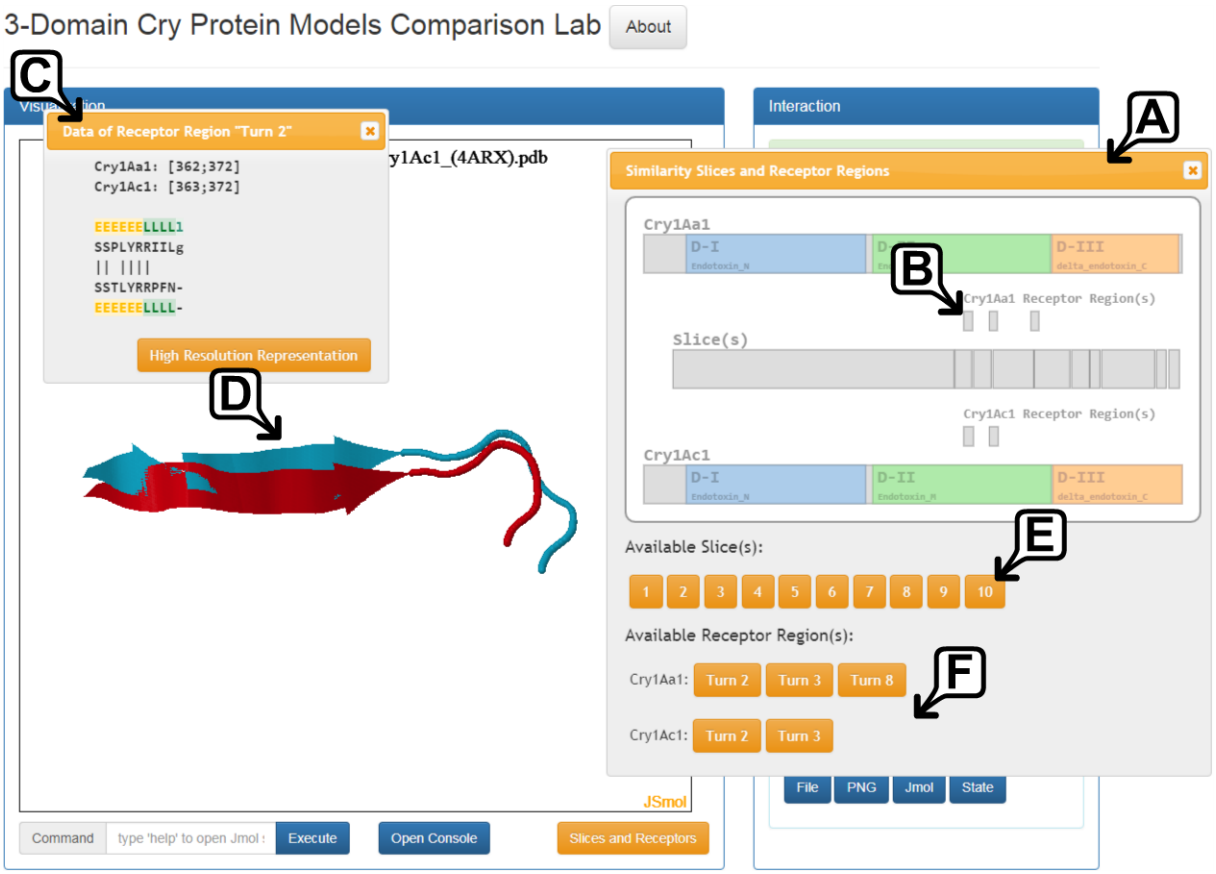
Figura 98 – Visão geral da interface gráfica principal da ferramenta "3-Domain Cry Protein Models Comparison Lab"



Fonte: Elaborada pelo autor

Os detalhes dos dados da comparação estrutural podem ser vistos na Figura 99. A seta "A" aponta para o painel intitulado "*Similarity Slices and Receptor Regions*", que por sua vez organiza esses detalhes em um diagrama. Nesse diagrama, as estruturas das proteínas Cry comparadas são exibidas de forma linear, além de serem destacadas as regiões similares que foram obtidas pelo algoritmo Dali bem como as regiões de interesse que foram analisadas nas Seções anteriores. A seta "B" aponta para uma dessas regiões, ou seja, a Volta 2 do Domínio II da proteína Cry1Aa1. Quando se clica nessa região, sua representação gráfica é exibida no painel de visualização (seta "D"), bem como seus detalhes estruturais são apresentados no painel indicado pela seta "C". Pode-se notar que essa região é a mesma que foi renderizada pelo *software* VMD e que foi exibida na Figura 90a. As setas "E" e "F" apontam para diversos botões que, ao serem clicados, funcionam da mesma forma que as regiões clicáveis apresentadas no diagrama apontado pela seta "A".

Figura 99 – Detalhe de uma região alinhada na ferramenta "3-Domain Cry Protein Models Comparison Lab".



Fonte: Elaborada pelo autor

O código fonte dessa ferramenta está disponibilizado no Apêndice C.

Pelo exposto, há indícios suficientes nos dados apresentados que corroboram para afirmar que existem modificações estruturais nas proteínas Cry que são responsáveis pela especificidade de cada uma delas. No próximo Capítulo serão apresentadas as conclusões desta pesquisa.

7 CONCLUSÕES

A partir do caminho percorrido na obtenção dos dados e, em seu tratamento para a geração dos resultados relativos à comparação estrutural entre as proteínas Cry de três domínios que foram utilizadas, pode-se inferir que pequenas modificações nas Volta 2 do Domínio II da proteína Cry1Aa1 não são capazes de influenciar sua especificidade quando comparada à proteína Cry1Ac1. Essa volta também não deve influenciar na especificidade com *Lepidoptera*, visto que há diferenças substanciais em relação à mesma região da proteína Cry2Aa1. A Volta 2 difere nas proteínas Cry3Aa1 e Cry3Bb1, ativas contra *Coleoptera*, indicando indícios que a mesma é importante para a toxicidade da proteína Cry1Aa1 nas ordens afetadas. Além disso, visto que as proteínas Cry4Aa1 e CryBa1, ambas ativas contra a ordem *Diptera*, não compartilham similaridade nessa região, pode-se também inferir que essa região, para a proteína Cry1Aa1, pode não conferir toxicidade de modo geral à ordem *Diptera*. As proteínas Cry5Ba1 e Cry8Ea1 não compartilham toxicidade com as mesmas ordens de Cry1Aa1, sendo assim, as diferenças encontradas nessa região provavelmente indicam a importância da mesma para a manifestação tóxica nas ordens afetadas pela proteína Cry1Aa1. A Volta 3 do Domínio II é similar apenas à mesma região da proteína Cry1Ac1, indicando ser importante para a especificidade nas ordens afetadas. Ainda em relação à proteína Cry1Aa1, a Volta 8 do Domínio II provavelmente confere toxicidade à essa proteína em algumas espécies das ordens afetadas, visto que essa região é substancialmente diferente na proteína Cry1Ac1, homóloga à Cry1Aa1. Em relação às Voltas 2 e 3 do Domínio II da proteína Cry1Ac1, pode-se afirmar que suas propriedades são as mesmas das apresentadas em relação à proteína Cry1Aa1.

A manifestação tóxica contra a ordem *Coleoptera* parece ter como região crucial o *Loop* 1 da proteína Cry3Aa1, visto que essa região é similar, principalmente em seus quatro primeiros resíduos, com as proteínas Cry3Bb1 e Cry8Ea1, também ativas contra *Coleoptera* e diferente quando comparada com as outras proteínas abordadas e que manifestam seu potencial tóxico a outras ordens que não a *Coleoptera*.

Ainda, os resíduos $^{157}\mathrm{R}^{159}$, $^{169}\mathrm{Y}^{171}$, $^{242}\mathrm{W}^{244}$, $^{245}\mathrm{F}^{247}$, $^{248}\mathrm{Y}^{250}$ e $^{263}\mathrm{F}^{265}$, todos contidos no Domínio I da proteína Cry4Ba1, parecem ter importância na manutenção da estabilidade da proteína e nos seus modos de ação gerais, visto que esses resíduos e suas regiões marginais são relativamente bem conservados em todas as proteínas verificadas.

Pelo exposto, pode-se concluir que as diferenças conformacionais nas estruturas das proteínas Cry, parecem contribuir para as suas manifestações tóxicas em diferentes ordens de insetos.

7.1 PUBLICAÇÃO

Durante o desenvolvimento desta tese foi publicado um artigo, no qual se descreve o processo de criação da ferramenta CryGetter: "BUZATTO, D.; FRANÇA, S. de C.; ZINGARETTI, S. M. CryGetter: a tool to automate retrieval and analysis of Cry protein data. *BMC Bioinformatics*, v. 17, n. 1, p. 1–14, 2016.".

7.2 *SOFTWARES* DESENVOLVIDOS

Como já relatado, foram desenvolvidos dois *softwares* durante o ciclo de vida desta tese, sendo eles o "CryGetter", um aplicativo capaz de consolidar dados das proteínas Cry a partir de duas fontes de dados e o "*3-Domain Cry Protein Models Comparison Lab*", utilizado para apresentar visualmente os resultados gerados neste trabalho.

7.3 TRABALHOS FUTUROS

É importante frisar que os resultados obtidos se encontram no espectro de simulação e comparação *in silico*, sendo necessário, como trabalho futuro, verificar se, ao modificar essas proteínas com técnicas de manipulação genética, as afirmações aqui apresentadas são sustentadas em modelos *in vivo*. Ainda, há a possibilidade de abordar a análise dos resultados obtidos nesta tese fazendo uso de técnicas de processamento de imagens ao invés da abordagem de inspeção visual a qual foi empregada.

REFERÊNCIAS

AGHILI, S.; AGRAWAL, D.; ABBADI, A. E. PADS: Protein Structure Alignment Using Directional Shape Signatures. *Database Systems for Advanced Applications*, v. 3453, p. 17–29, 2005.

ANGELO, E. A.; VILAS-BÔAS, G. T.; CASTRO-GÓMEZ, R. J. H. Bacillus thuringiensis: características gerais e fermentação. *Ciências Agrárias*, v. 31, p. 945–958, 2010.

ASHBY, C. et al. New enumeration algorithm for protein structure comparison and classification. *BMC genomics*, v. 14, 2013.

AUNG, Z.; TAN, K.-L. MatAlign: Precise Protein Structure Comparison by Matrix Alignment. *Journal of Bioinformatics and Computational Biology*, v. 04, n. 06, p. 1197–1216, 2006.

BARTON, G. J.; STERNBERG, M. J. A strategy for the rapid multiple alignment of protein sequences. Confidence levels from tertiary structure comparisons. *J Mol Biol*, v. 198, n. 2, p. 327–337, 1987.

BERMAN, H. M. et al. The Protein Data Bank. *Nucleic Acids Research*, v. 28, n. 1, p. 235–242, 2000.

BOONSERM, P.; ANGSUTHANASOMBAT, C.; LESCAR, J. Crystallization and preliminary crystallographic study of the functional form of the Bacillus thuringiensis mosquito-larvicidal Cry4Aa mutant toxin. *Acta Crystallogr. D Biol. Crystallogr.*, v. 60, n. Pt 7, p. 1315–1318, 2004.

BOONSERM, P. et al. Crystal structure of the mosquito-larvicidal toxin Cry4Ba and its biological implications. *J. Mol. Biol.*, v. 348, n. 2, p. 363–382, 2005.

BOONSERM, P. et al. Structure of the functional form of the mosquito larvicidal Cry4Aa toxin from Bacillus thuringiensis at a 2.8-angstrom resolution. *J. Bacteriol.*, v. 188, n. 9, p. 3391–3401, 2006.

BRANDEN, C.; TOOZE, J. *Introduction to Protein Structure*. 2. ed. New York: Garland Science, 1999. 410 p.

BRAVO, A.; GILL, S. S.; SOBERON, M. Mode of action of Bacillus thuringiensis Cry and Cyt toxins and their potential for insect control. *Toxicon*, v. 49, n. 4, p. 423–435, 2007.

BRAVO, A. et al. Oligomerization triggers binding of a Bacillus thuringiensis Cry1Ab pore-forming toxin to aminopeptidase N receptor leading to insertion into membrane microdomains. *Biochim Biophys Acta*, v. 1667, n. 1, p. 38–46, 2004.

BRETSCHNEIDER, A.; HECKEL, D. G.; PAUCHET, Y. Three toxins, two receptors, one mechanism: Mode of action of Cry1A toxins from Bacillus thuringiensis in Heliothis virescens. *Insect Biochem Mol Biol*, 2016.

BUSS, D. S.; CALLAGHAN, A. Interaction of pesticides with p-glycoprotein and other ABC proteins: A survey of the possible importance to insecticide, herbicide and fungicide resistance. *Pesticide Biochemistry and Physiology*, v. 90, n. 3, p. 141–153, 2008.

BUZATTO, D.; FRANÇA, S. de C.; ZINGARETTI, S. M. CryGetter: a tool to automate retrieval and analysis of Cry protein data. *BMC Bioinformatics*, v. 17, n. 1, p. 1–14, 2016.

CAN, T.; WANG, Y. F. Protein structure alignment and fast similarity search using local shape signatures. *J Bioinform Comput Biol*, v. 2, n. 1, p. 215–239, 2004.

CARUGO, O.; PONGOR, S. A normalized root-mean-spuare distance for comparing protein three-dimensional structures. *Protein Sci*, v. 10, n. 7, p. 1470–1473, 2001.

CASTRIGNANO, T. et al. The PMDB Protein Model Database. *Nucleic Acids Res*, v. 34, n. Database issue, p. D306–9, 2006.

CHIONH, C.-H. et al. Towards Scaleable Protein Structure Comparison and Database Search. *International Journal on Artificial Intelligence Tools*, v. 14, n. 05, p. 827–848, 2005.

CHOTHIA, C.; FINKELSTEIN, A. V. The classification and origins of protein folding patterns. *Annu Rev Biochem*, v. 59, p. 1007–1039, 1990.

CHU, C.-h. et al. Angle-distance image matching techniques for protein structure comparison. *Journal of Molecular Recognition*, v. 21, n. 6, p. 442–452, 2008.

CIRIELLO, G.; COMIN, M.; GUERRA, C. Algorithmic re-structuring and data replication for protein structure comparison on a GRID. *Future Generation Computer Systems*, v. 23, n. 3, p. 391–397, 2007.

COMIN, M.; GUERRA, C.; ZANOTTI, G. PROuST: A Comparison Method of Three-Dimensional Structures of Proteins Using Indexing Techniques. *Journal of Computational Biology*, v. 11, n. 6, p. 1061–1072, 2004.

CRICK, F. Central dogma of molecular biology. *Nature*, v. 227, n. 5258, p. 561–563, 1970.

CRICK, F. H. On protein synthesis. *Symp Soc Exp Biol*, v. 12, p. 138–163, 1958.

CRICKMORE, N. et al. *Bacillus thuringiensis toxin nomenclature*. 2016. Disponível em: <http://www.btnomenclature.info>. Acesso em: 04 de outubro de 2016.

CRICKMORE, N. et al. Revision of the nomenclature for the Bacillus thuringiensis pesticidal crystal proteins. *Microbiol. Mol. Biol. Rev.*, v. 62, n. 3, p. 807–813, 1998.

CRYSTAL protein Bacillus thuringiensis - Protein - NCBI. 2016. Disponível em: <https://www.ncbi.nlm.nih.gov/protein?cmd=Retrieve&dopt=GenPept&list_uids=142765>. Acesso em: 04 de outubro de 2016.

CSABA, G.; BIRZELE, F.; ZIMMER, R. Protein structure alignment considering phenotypic plasticity. *Bioinformatics*, v. 24, n. 16, p. 98–104, 2008.

DA-FU, D.; JIANG, Q.; ZU-KANG, F. A differential geometric treatment of protein structure comparison. *Bulletin of Mathematical Biology*, v. 56, n. 5, p. 923–943, 1994.

DAYHOFF, M. O.; SCHWARTZ, R. M.; ORCUTT, B. C. Chapter 22: A model of evolutionary change in proteins. In: _____. *Atlas of Protein Sequence and Structure*. Washington DC: National Biomedical Research Foundation, 1978. v. 5, p. 345–352.

DEHURY, B. et al. Structural analysis and molecular dynamics simulations of novel delta-endotoxin Cry1Id from Bacillus thuringiensis to pave the way for development of novel fusion proteins against insect pests of crops. *J Mol Model*, v. 19, n. 12, p. 5301–5316, 2013.

DEMENTIEV, A. et al. The pesticidal Cry6Aa toxin from Bacillus thuringiensis is structurally similar to HlyE-family alpha pore-forming toxins. *BMC Biol*, v. 14, p. 71, 2016.

DERBYSHIRE, D. J.; ELLAR, D. J.; LI, J. Crystallization of the Bacillus thuringiensis toxin Cry1Ac and its complex with the receptor ligand N-acetyl-D-galactosamine. *Acta Crystallogr. D Biol. Crystallogr.*, v. 57, n. Pt 12, p. 1938–1944, 2001.

EARGLE, J.; WRIGHT, D.; LUTHEY-SCHULTEN, Z. Multiple Alignment of protein structures and sequences for VMD. *Bioinformatics*, v. 22, n. 4, p. 504–506, 2006.

ELLEUCH, J. et al. Cry4Ba and Cyt1Aa proteins from Bacillus thuringiensis israelensis: Interactions and toxicity mechanism against Aedes aegypti. *Toxicon*, v. 104, p. 83–90, 2015.

ENDO, H. et al. Cry toxin specificities of insect ABCC transporters closely related to lepidopteran ABCC2 transporters. *Peptides*, 2017.

ESLAHCHI, C. et al. STON: A novel method for protein three-dimensional structure comparison. *Computers in Biology and Medicine*, v. 39, n. 2, p. 166–172, 2009.

FENG, D. et al. Domain III of Bacillus thuringiensis Cry1Ie Toxin Plays an Important Role in Binding to Peritrophic Membrane of Asian Corn Borer. *PLoS One*, v. 10, n. 8, p. e0136430, 2015.

FERRARI, C.; GUERRA, C.; ZANOTTI, G. A grid-aware approach to protein structure comparison. *Journal of Parallel and Distributed Computing*, v. 63, n. 7, p. 728–737, 2003.

FIÚZA, L. M.; BERLITZ, D. L. Produtos de Bacillus Thuringiensis Registro e Comercialização. *Biotecnologia, Ciência e Desenvolvimento*, v. 38, p. 58–60, 2009.

FIÚZA, L. M.; PINTO, L. M. N. Plantas Transgênicas que Sintetizam Toxinas de Bacillus Thuringiensis e Outras. *Biotecnologia, Ciência e Desenvolvimento*, v. 38, p. 62–67, 2009.

FRANCIS, B. R.; BULLA, L. A. Further characterization of BT-R1, the cadherin-like receptor for Cry1Ab toxin in tobacco hornworm (Manduca sexta) midguts. *Insect Biochem. Mol. Biol.*, v. 27, n. 6, p. 541–550, 1997.

GALGONEK, J.; HOKSZA, D.; SKOPAL, T. SProt: sphere-based protein structure similarity algorithm. *Proteome Sci*, v. 9 Suppl 1, p. S20, 2011.

GALITSKY, N. et al. Structure of the insecticidal bacterial delta-endotoxin Cry3Bb1 of Bacillus thuringiensis. *Acta Crystallogr. D Biol. Crystallogr.*, v. 57, n. Pt 8, p. 1101–1109, 2001.

GELLY, J.-C. et al. iPBA: a tool for protein structure comparison using sequence alignment strategies. *Nucleic Acids Research*, v. 39, n. suppl_2, p. W18–W23, 2011.

GIBRAT, J.-F.; MADEJ, T.; BRYANT, S. H. Surprising similarities in structure comparison. *Current Opinion in Structural Biology*, v. 6, n. 3, p. 377–385, 1996.

GOMEZ, I. et al. Specific epitopes of domains II and III of Bacillus thuringiensis Cry1Ab toxin involved in the sequential interaction with cadherin and aminopeptidase-N receptors in Manduca sexta. *J Biol Chem*, v. 281, n. 45, p. 34032–34039, 2006.

GOMEZ, I. et al. Hydropathic complementarity determines interaction of epitope (869)HITDTNNK(876) in Manduca sexta Bt-R(1) receptor with loop 2 of domain II of Bacillus thuringiensis Cry1A toxins. *J. Biol. Chem.*, v. 277, n. 33, p. 30137–30143, 2002.

GOMEZ, I. et al. Role of receptor interaction in the mode of action of insecticidal Cry and Cyt toxins produced by Bacillus thuringiensis. *Peptides*, v. 28, n. 1, p. 169–173, 2007.

GOMEZ, I. et al. Cadherin-like receptor binding facilitates proteolytic cleavage of helix alpha-1 in domain I and oligomer pre-pore formation of Bacillus thuringiensis Cry1Ab toxin. *FEBS Lett*, v. 513, n. 2-3, p. 242–246, 2002.

GOWDA, A. et al. A transgenic approach for controlling Lygus in cotton. *Nat Commun*, v. 7, p. 12213, 2016.

GROCHULSKI, P. et al. Bacillus thuringiensis CryIA(a) insecticidal toxin: crystal structure and channel formation. *J. Mol. Biol.*, v. 254, n. 3, p. 447–464, 1995.

GUDA, C. et al. CE-MC: a multiple protein structure alignment server. *Nucleic Acids Res*, v. 32, n. Web Server issue, p. W100–3, 2004.

GUERLER, A.; KNAPP, E.-W. GIS: a comprehensive source for protein structure similarities. *Nucleic Acids Research*, v. 38, n. Web Server issue, p. W46–W52, 2010.

GUEX, N.; PEITSCH, M. C. SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. *Electrophoresis*, v. 18, n. 15, p. 2714–2723, 1997.

GUO, S. et al. Crystal structure of Bacillus thuringiensis Cry8Ea1: An insecticidal toxin toxic to underground pests, the larvae of Holotrichia parallela. *J. Struct. Biol.*, v. 168, n. 2, p. 259–266, 2009.

GUTIERREZ, F. I. et al. Efficient and automated large-scale detection of structural relationships in proteins with a flexible aligner. *BMC Bioinformatics*, v. 17, n. 1, p. 20, 2016.

GUTIERREZ, P.; ALZATE, O.; ORDUZ, S. A theoretical model of the tridimensional structure of Bacillus thuringiensis subsp. medellin Cry 11Bb toxin deduced by homology modelling. *Mem. Inst. Oswaldo Cruz*, v. 96, n. 3, p. 357–364, 2001.

HASEGAWA, H.; HOLM, L. Advances and pitfalls of protein structural alignment. *Current Opinion in Structural Biology*, v. 19, n. 3, p. 341–348, 2009.

HENIKOFF, S.; HENIKOFF, J. G. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, v. 89, n. 22, p. 10915–10919, 1992.

HOLM, L.; LAAKSO, L. M. Dali server update. *Nucleic Acids Res*, 2016.

HOLM, L.; PARK, J. DaliLite workbench for protein structure comparison. *Bioinformatics*, v. 16, n. 6, p. 566–567, 2000.

HOLM, L.; ROSENSTROM, P. Dali server: conservation mapping in 3D. *Nucleic Acids Res*, v. 38, n. Web Server issue, p. W545–9, 2010.

HOLM, L.; SANDER, C. Protein Structure Comparison by Alignment of Distance Matrices. *Journal of Molecular Biology*, v. 233, n. 1, p. 123–138, 1993.

HOLM, L.; SANDER, C. Dali: a network tool for protein structure comparison. *Trends in Biochemical Sciences*, v. 20, n. 11, p. 478–480, 1995.

How does Bt work. 2016. Disponível em: <http://www.bt.ucsd.edu/how_bt_work.html>. Acesso em: 19 de agosto de 2016.

HUANG, J. et al. Crystal structure of Cry6Aa: A novel nematicidal ClyA-type alpha-pore-forming toxin from Bacillus thuringiensis. *Biochem Biophys Res Commun*, v. 478, n. 1, p. 307–313, 2016.

HUI, F. et al. Structure and glycolipid binding properties of the nematicidal protein Cry5B. *Biochemistry*, v. 51, n. 49, p. 9911–9921, 2012.

HUMPHREY, W.; DALKE, A.; SCHULTEN, K. VMD - Visual Molecular Dynamics. *Journal of Molecular Graphics*, v. 14, p. 33–38, 1996.

HWANG, K. Y. et al. Structure-based identification of a novel NTPase from Methanococcus jannaschii. *Nat Struct Biol*, v. 6, n. 7, p. 691–696, 1999.

IAKOVIDOU, N. et al. Going over the three dimensional protein structure similarity problem. *Artificial Intelligence Review*, v. 42, n. 3, p. 445–459, 2014.

JIA, Y. et al. A new scoring function and associated statistical significance for structure alignment by CE. *J Comput Biol*, v. 11, n. 5, p. 787–799, 2004.

JIN, T. et al. Identification of an alkaline phosphatase as a putative Cry1Ac binding protein in Ostrinia furnacalis (Guenee). *Pestic Biochem Physiol*, v. 131, p. 80–86, 2016.

JMOL: an open-source Java viewer for chemical structures in 3D. 2016. Disponível em: <http://www.jmol.org/>. Acesso em: 25 de outubro de 2016.

JOSEPH, A. P.; SRINIVASAN, N.; BREVERN, A. G. de. Improvement of protein structure comparison using a structural alphabet. *Biochimie*, v. 93, n. 9, p. 1434–1445, 2011.

JOSEPH, A. P.; SRINIVASAN, N.; BREVERN, A. G. de. Progressive structure-based alignment of homologous proteins: Adopting sequence comparison strategies. *Biochimie*, v. 94, n. 9, p. 2025–2034, 2012.

JUNG, S. et al. Protein backbone torsion angle-based structure comparison and secondary structure database web server. *Genomics & informatics*, v. 11, n. 3, p. 155, 2013.

KASHYAP, S. Computational Modeling Deduced Three Dimensional Structure of Cry1Ab16 Toxin from Bacillus thuringiensis AC11. *Indian J. Microbiol.*, v. 52, n. 2, p. 263–269, 2012.

KASHYAP, S.; SINGH, B. D.; AMLA, D. V. Computational tridimensional protein modeling of Cry1Ab19 toxin from Bacillus thuringiensis BtX-2. *J. Microbiol. Biotechnol.*, v. 22, n. 6, p. 788–792, 2012.

KAWABATA, T. MATRAS: a program for protein 3D structure comparison. *Nucleic Acids Research*, v. 31, n. 13, p. 3367–3369, 2003.

KAWABATA, T.; NISHIKAWA, K. Protein structure comparison using the markov transition model of evolution. *Proteins*, v. 41, n. 1, p. 108–122, 2000.

KEETON, T. P.; BULLA, L. A. Ligand specificity and affinity of BT-R1, the Bacillus thuringiensis Cry1A toxin receptor from Manduca sexta, expressed in mammalian and insect cell cultures. *Appl. Environ. Microbiol.*, v. 63, n. 9, p. 3419–3425, 1997.

KELKER, M. S. et al. Structural and biophysical characterization of Bacillus thuringiensis insecticidal proteins Cry34Ab1 and Cry35Ab1. *PLoS ONE*, v. 9, n. 11, p. e112555, 2014.

KESSEL, A.; BEN-TAL, N. *Introduction to Proteins: Structure, Function, and Motion*. Boca Raton: CRC Press, 2011. 591 p.

KIFER, I.; NUSSINOV, R.; WOLFSON, H. J. GOSSIP: a method for fast and accurate global alignment of protein structures. *Bioinformatics*, v. 27, n. 7, p. 925–932, 2011.

KNOWLES, B. H.; DOW, J. A. T. The Crystal delta-endotoxins of Bacillus thuringiensis: Models for their Mechanism of Action on the Insect Gut. *Bioessays*, v. 15, n. 7, p. 469–476, 1993.

KNOWLES, B. H.; ELLAR, D. J. Colloid-osmotic lysis is a general feature of the mechanism of action of Bacillus thuringiensis d-endotoxins with different insect specificity. *Biochimica et Biophysica Acta (BBA) - General Subjects*, v. 924, n. 3, p. 509–518, 1987.

KOCH, M. S. et al. The food and environmental safety of Bt crops. *Front Plant Sci*, v. 6, p. 283, 2015.

KOEHL, P. Protein structure similarities. *Current Opinion in Structural Biology*, v. 11, n. 3, p. 348–353, 2001.

KOTLOVYI, V.; NICHOLS, W. L.; EYCK, L. F. T. Protein structural alignment for detection of maximally conserved regions. *Biophysical Chemistry*, v. 105, n. 2, p. 595–608, 2003.

LATHROP, R. H. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng*, v. 7, n. 9, p. 1059–1068, 1994.

LEAST Squares Fitting – from Wolfram MathWorld. 2016. Disponível em: <http://mathworld.wolfram.com/LeastSquaresFitting.html>. Acesso em: 25 de abril de 2016.

LEBEL, G. et al. Mutations in domain I interhelical loops affect the rate of pore formation by the Bacillus thuringiensis Cry1Aa toxin in insect midgut brush border membrane vesicles. *Appl Environ Microbiol*, v. 75, n. 12, p. 3842–3850, 2009.

LEETACHEWA, S. et al. Functional characterizations of residues Arg-158 and Tyr-170 of the mosquito-larvicidal Bacillus thuringiensis Cry4Ba. *BMB Rep*, v. 47, n. 10, p. 546–551, 2014.

LESK, A. M. *Introduction to Protein Science: Architecture, Function and Genomics*. 3. ed. United Kingdom: Oxford University Press, 2016. 488 p.

LI, J. D.; CARROLL, J.; ELLAR, D. J. Crystal structure of insecticidal delta-endotoxin from Bacillus thuringiensis at 2.5 A resolution. *Nature*, v. 353, n. 6347, p. 815–821, 1991.

LI, S. C.; NG, Y. K. On protein structure alignment under distance constraint. *Theoretical Computer Science*, v. 412, n. 32, p. 4187–4199, 2011.

LIU, W.; SRIVASTAVA, A.; ZHANG, J. A Mathematical Framework for Protein Structure Comparison. *PLoS Computational Biology*, v. 7, n. 2, p. e1001075, 2011.

MAAGD, R. A. de; BRAVO, A.; CRICKMORE, N. How Bacillus thuringiensis has evolved specific toxins to colonize the insect world. *Trends Genet.*, v. 17, n. 4, p. 193–199, 2001.

MADEJ, T. et al. MMDB and VAST+: tracking structural similarities between macromolecular complexes. *Nucleic Acids Res*, v. 42, n. Database issue, p. D297–303, 2014.

MAIOROV, V. N.; CRIPPEN, G. M. Significance of Root-Mean-Square Deviation in Comparing Three-dimensional Structures of Globular Proteins. *Journal of Molecular Biology*, v. 235, n. 2, p. 625–634, 1994.

MALOD-DOGNIN, N.; ANDONOV, R.; YANEV, N. Maximum Cliques in Protein Structure Comparison. 2009.

MARTINEZ-RAMIREZ, A. C. et al. Ligand blot identification of a Manduca sexta midgut binding protein specific to three Bacillus thuringiensis CryIA-type ICPs. *Biochem Biophys Res Commun*, v. 201, n. 2, p. 782–787, 1994.

MATHEWS, C. et al. *Biochemistry*. 4. ed. [S.l.]: Pearson Education, 2012.

MERNBERGER, M.; KLEBE, G.; HULLERMEIER, E. SEGA: Semiglobal Graph Alignment for Structure-Based Protein Comparison. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, v. 8, n. 5, p. 1330–1343, 2011.

MIRANDA, R.; ZAMUDIO, F. Z.; BRAVO, A. Processing of Cry1Ab delta-endotoxin from Bacillus thuringiensis by Manduca sexta and Spodoptera frugiperda midgut proteases: role in protoxin activation and toxin inactivation. *Insect Biochem Mol Biol*, v. 31, n. 12, p. 1155–1163, 2001.

MLA CE Course Manual: Molecular Biology Information Resources (Entrez). 2016. Disponível em: <https://www.ncbi.nlm.nih.gov/Class/MLACourse/Original8Hour/Entrez/>. Acesso em: 04 de outubro de 2016.

MORSE, R. J.; YAMAMOTO, T.; STROUD, R. M. Structure of Cry2Aa suggests an unexpected receptor binding epitope. *Structure*, v. 9, n. 5, p. 409–417, 2001.

MOULT, J. et al. Critical assessment of methods of protein structure prediction (CASP): round III. *Proteins*, Suppl 3, p. 2–6, 1999.

MURZIN, A. G. Structural classification of proteins: new superfamilies. *Curr Opin Struct Biol*, v. 6, n. 3, p. 386–394, 1996.

MURZIN, A. G. How far divergent evolution goes in proteins. *Curr Opin Struct Biol*, v. 8, n. 3, p. 380–387, 1998.

MURZIN, A. G. et al. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, v. 247, n. 4, p. 536–540, 1995.

NEEDLEMAN, S. B.; WUNSCH, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, v. 48, n. 3, p. 443–453, 1970.

NELSON, D. L.; COX, M. M. *Princípios de Bioquímica de Lehninger*. 6. ed. Porto Alegre: Artmed, 2014. 1328 p.

ORTIZ, A. R.; STRAUSS, C. E.; OLMEA, O. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci*, v. 11, n. 11, p. 2606–2621, 2002.

PALMA, L. et al. Bacillus thuringiensis toxins: an overview of their biocidal activity. *Toxins (Basel)*, v. 6, n. 12, p. 3296–3325, 2014.

PELTA, D. et al. A fuzzy sets based generalization of contact maps for the overlap of protein structures. *Fuzzy Sets and Systems*, v. 152, n. 1, p. 103–123, 2005.

PELTA, D. A.; GONZÁLEZ, J. R.; VEGA, M. M. A simple and fast heuristic for protein structure comparison. *BMC Bioinformatics*, v. 9, p. 161–161, 2008.

PINTO, L. M. N. et al. Toxinas de Bacillus Thuringiensis. *Biotecnologia, Ciência e Desenvolvimento*, v. 38, p. 24–31, 2009.

PRLIC, A. et al. BioJava: an open-source framework for bioinformatics in 2012. *Bioinformatics*, v. 28, n. 20, p. 2693–2695, 2012.

QIU, L. et al. Cadherin is involved in the action of Bacillus thuringiensis toxins Cry1Ac and Cry2Aa in the beet armyworm, Spodoptera exigua. *J. Invertebr. Pathol.*, v. 127, p. 47–53, 2015.

RAYMOND, B.; FEDERICI, B. A. In defense of Bacillus thuringiensis, the safest and most successful microbial insecticide available to humanity - a response to EFSA. *FEMS Microbiol Ecol*, 2017.

RAZMARA, J.; DERIS, S.; PARVIZPOUR, S. TS-AMIR: a topology string alignment method for intensive rapid protein structure comparison. *Algorithms for Molecular Biology*, v. 7, n. 1, p. 4–4, 2012.

RAZMARA, J.; PARVIZPOUR, S.; SAMIRA, F. Flexible Protein Structure Alignment Based on Topology String Alignment of Secondary Structure. *International Journal of e-Education, e-Business, e-Management and e-Learning*, v. 4, n. 1, p. 19–22, 2014.

ROBERTS, E. et al. MultiSeq: unifying sequence and structure data for evolutionary analysis. *BMC Bioinformatics*, v. 7, p. 382, 2006.

ROH, J. Y. et al. Bacillus thuringiensis as a specific, safe, and effective tool for insect pest control. *J. Microbiol. Biotechnol.*, v. 17, n. 4, p. 547–559, 2007.

RUSSELL, R. B.; BARTON, G. J. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins*, v. 14, n. 2, p. 309–323, 1992.

SANAHUJA, G. et al. Bacillus thuringiensis: a century of research, development and commercial applications. *Plant Biotechnol. J.*, v. 9, n. 3, p. 283–300, 2011.

SASIN, J. M.; KUROWSKI, M. A.; BUJNICKI, J. M. STRUCLA: a WWW meta-server for protein structure comparison and evolutionary classification. *Bioinformatics*, v. 19, n. 1, p. 252–254, 2003.

SAWAYA, M. R. et al. Protein crystal structure obtained at 2.9 A resolution from injecting bacterial cells into an X-ray free-electron laser beam. *Proc. Natl. Acad. Sci. U.S.A.*, v. 111, n. 35, p. 12769–12774, 2014.

SCHNEPF, H. E.; WHITELEY, H. R. Cloning and expression of the Bacillus thuringiensis crystal protein gene in Escherichia coli. *Proc. Natl. Acad. Sci. U.S.A.*, v. 78, n. 5, p. 2893–2897, 1981.

SCHNEPF, H. E.; WONG, H. C.; WHITELEY, H. R. The amino acid sequence of a crystal protein from Bacillus thuringiensis deduced from the DNA base sequence. *J Biol Chem*, v. 260, n. 10, p. 6264–6272, 1985.

SCHRODINGER, L. The PyMOL Molecular Graphics System, Version 1.8. 2015.

SCOTT, G. J.; SHYU, C.-R. EBS k-d Tree: An Entropy Balanced Statistical k-d Tree for Image Databases with Ground-Truth Labels. In: _____. *Image and Video Retrieval: Second International Conference, CIVR 2003 Urbana-Champaign, IL, USA, July 24–25, 2003 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. p. 467–477. ISBN 978-3-540-45113-6.

SHAPIRO, J.; BRUTLAG, D. FoldMiner and LOCK 2: protein structure comparison and motif discovery on the web. *Nucleic acids research*, v. 32, n. Web Server issue, p. W536, 2004.

SHAPIRO, J.; BRUTLAG, D. FoldMiner: Structural motif discovery using an improved superposition algorithm. *Protein Sci*, v. 13, n. 1, p. 278–294, 2004.

SHARMA, A.; MANOLAKOS, E. S. Efficient Multicriteria Protein Structure Comparison on Modern Processor Architectures. *Biomed Res Int*, v. 2015, p. 563674, 2015.

SHARMA, A.; PAPANIKOLAOU, A.; MANOLAKOS, E. S. Accelerating All-to-All Protein Structures Comparison with TMalign Using a NoC Many-Cores Processor Architecture. In: *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*. [S.l.: s.n.], 2013. p. 510–519.

SHATSKY, M.; NUSSINOV, R.; WOLFSON, H. J. Flexible protein alignment and hinge detection. *Proteins*, v. 48, n. 2, p. 242–256, 2002.

SHIH, E. S. C.; HWANG, M.-J. Protein structure comparison by probability-based matching of secondary structure elements. *Bioinformatics*, v. 19, n. 6, p. 735–741, 2003.

SHINDYALOV, I. N.; BOURNE, P. E. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, v. 11, n. 9, p. 739–747, 1998.

SHINDYALOV, I. N.; BOURNE, P. E. A database and tools for 3-D protein structure comparison and alignment using the Combinatorial Extension (CE) algorithm. *Nucleic acids research*, v. 29, n. 1, p. 228, 2001.

SHU, C. et al. Assembling of Holotrichia parallela (dark black chafer) midgut tissue transcriptome and identification of midgut proteins that bind to Cry8Ea toxin from Bacillus thuringiensis. *Appl Microbiol Biotechnol*, v. 99, n. 17, p. 7209–7218, 2015.

SILLITOE, I. et al. CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res*, v. 43, n. Database issue, p. D376–81, 2015.

SIPPL, M. J. On distance and similarity in fold space. *Bioinformatics*, v. 24, n. 6, p. 872–873, 2008.

SIPPL, M. J. et al. A discrete view on fold space. *Bioinformatics*, v. 24, n. 6, p. 870–871, 2008.

SIPPL, M. J.; WIEDERSTEIN, M. A note on difficult structure alignment problems. *Bioinformatics*, v. 24, n. 3, p. 426–427, 2008.

SMITH, T. F.; WATERMAN, M. S. Identification of common molecular subsequences. *J Mol Biol*, v. 147, n. 1, p. 195–197, 1981.

SRIWIMOL, W. et al. Potential Prepore Trimer Formation by the Bacillus thuringiensis Mosquito-specific Toxin: MOLECULAR INSIGHTS INTO A CRITICAL PREREQUISITE OF MEMBRANE-BOUND MONOMERS. *J Biol Chem*, v. 290, n. 34, p. 20793–20803, 2015.

SUYAMA, M.; MATSUO, Y.; NISHIKAWA, K. Comparison of protein structures using 3D profile alignment. *Journal of Molecular Evolution*, v. 44, n. 1, p. S163–S173, 1997.

TAYLOR, W. R. Protein structure comparison using iterated double dynamic programming. *Protein Sci.*, v. 8, n. 3, p. 654–665, 1999.

TAYLOR, W. R. Protein structure comparison using SAP. *Methods Mol Biol*, v. 143, p. 19–32, 2000.

TAYLOR, W. R. Protein structure comparison using bipartite graph matching and its application to protein structure classification. *Mol. Cell Proteomics*, v. 1, n. 4, p. 334–339, 2002.

TERASHI, G.; TAKEDA-SHITAKA, M. CAB-Align: A Flexible Protein Structure Alignment Method Based on the Residue-Residue Contact Area. *PLoS One*, v. 10, n. 10, p. e0141440, 2015.

THOMPSON, J. D.; HIGGINS, D. G.; GIBSON, T. J. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, v. 22, n. 22, p. 4673–4680, 1994.

TIEWSIRI, K.; ANGSUTHANASOMBAT, C. Structurally conserved aromaticity of Tyr249 and Phe264 in helix 7 is important for toxicity of the Bacillus thuringiensis Cry4Ba toxin. *J Biochem Mol Biol*, v. 40, n. 2, p. 163–171, 2007.

TRIOLA, M. F. *Introdução à Estatística: atualização da tecnologia*. 11. ed. Rio de Janeiro: LTC, 2014. 707 p.

TYAGI, M. et al. Protein structure mining using a structural alphabet. *Proteins*, v. 71, n. 2, p. 920–937, 2008.

VADLAMUDI, R. K.; JI, T. H.; BULLA L. A., J. A specific binding protein from Manduca sexta for the insecticidal toxin of Bacillus thuringiensis subsp. berliner. *J Biol Chem*, v. 268, n. 17, p. 12334–12340, 1993.

VESTERSTROEM, J.; TAYLOR, W. Flexible Secondary Structure Based Protein Structure Comparison Applied to the Detection of Circular Permutation. *Journal of Computational Biology*, v. 13, n. 1, p. 43–62, 2006.

WANG, S. et al. Protein structure alignment beyond spatial proximity. *Sci Rep*, v. 3, p. 1448, 2013.

WRABL, J. O.; GRISHIN, N. V. Statistics of random protein superpositions: p-values for pairwise structure alignment. *J Comput Biol*, v. 15, n. 3, p. 317–355, 2008.

WU, Z. et al. A new geometric-topological method to measure protein fold similarity. *Chemical Physics Letters*, v. 433, n. 4, p. 432–438, 2007.

WWPDB: X-ray validation report user guide. 2016. Disponível em: <http: //wwpdb.org/validation/legacy/XrayValidationReportHelp>. Acesso em: 18 de outubro de 2016.

XIA, L. Q. et al. The theoretical 3D structure of Bacillus thuringiensis Cry5Ba. *J Mol Model*, v. 14, n. 9, p. 843–848, 2008.

XIN-MIN, Z. et al. The theoretical three-dimensional structure of Bacillus thuringiensis Cry5Aa and its biological implications. *Protein J.*, v. 28, n. 2, p. 104–110, 2009.

XU, C. et al. Crystal structure of Cry51Aa1: A potential novel insecticidal aerolysin-type β-pore-forming toxin from Bacillus thuringiensis. *Biochem. Biophys. Res. Commun.*, v. 462, n. 3, p. 184–189, 2015.

XU, C. et al. Structural insights into Bacillus thuringiensis Cry, Cyt and parasporin toxins. *Toxins (Basel)*, v. 6, n. 9, p. 2732–2770, 2014.

XU, D. et al. ProteinDBS: A real-time retrieval system for protein structure comparison. *Nucleic Acids Research*, v. 32, p. W527–W575, 2004.

YE, Y.; GODZIK, A. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, v. 19 Suppl 2, p. ii246–55, 2003.

YE, Y.; GODZIK, A. FATCAT: a web server for flexible structure comparison and structure similarity searching. *Nucleic Acids Research*, v. 32, n. Web Server issue, p. W582–W585, 2004.

YONA, G.; KEDEM, K. The URMS-RMS hybrid algorithm for fast and sensitive local protein structure alignment. *J. Comput. Biol.*, v. 12, n. 1, p. 12–32, 2005.

ZHANG, H. et al. Cadherin mutation linked to resistance to Cry1Ac affects male paternity and sperm competition in Helicoverpa armigera. *J. Insect Physiol.*, v. 70, p. 67–72, 2014.

ZHANG, H. et al. Non-recessive Bt toxin resistance conferred by an intracellular cadherin mutation in field-selected populations of cotton bollworm. *PLoS One*, v. 7, n. 12, p. e53418, 2012.

ZHANG, L. et al. A fast indexing approach for protein structure comparison. *BMC bioinformatics*, v. 11 Suppl 1, p. S46, 2010.

ZHANG, Q.; HUA, G.; ADANG, M. J. Chitosan/DsiRNA nanoparticle targeting identifies AgCad1 cadherin in Anopheles gambiae larvae as an in vivo receptor of Cry11Ba toxin of Bacillus thuringiensis subsp. jegathesan. *Insect Biochem. Mol. Biol.*, v. 60, p. 33–38, 2015.

ZHANG, X. et al. Cytotoxicity of Bacillus thuringiensis Cry1Ab toxin depends on specific binding of the toxin to the cadherin receptor BT-R1 expressed in insect cells. *Cell Death Differ*, v. 12, n. 11, p. 1407–1416, 2005.

ZHANG, Y.; SKOLNICK, J. Scoring function for automated assessment of protein structure template quality. *Proteins*, v. 57, n. 4, p. 702–710, 2004.

ZHANG, Y.; SKOLNICK, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*, v. 33, n. 7, p. 2302–2309, 2005.

ZHAO, C.; SACAN, A. UniAlign: protein structure alignment meets evolution. *Bioinformatics*, 2015.

ZHAO, X. M.; ZHOU, P. D.; XIA, L. Q. Homology modeling of mosquitocidal Cry30Ca2 of Bacillus thuringiensis and its molecular docking with N-acetylgalactosamine. *Biomed. Environ. Sci.*, v. 25, n. 5, p. 590–596, 2012.

ZOTENKO, E. et al. Structural footprinting in protein structure comparison: the impact of structural fragments. *BMC Structural Biology*, v. 7, n. 53, p. 53, 2007.

ZOTENKO, E.; O'LEARY, D. P.; PRZYTYCKA, T. M. Secondary structure spatial conformation footprint: a novel method for fast protein structure comparison and classification. *BMC Structural Biology*, v. 6, p. 12–12, 2006.

ZUNIGA-NAVARRETE, F. et al. Identification of Bacillus thuringiensis Cry3Aa toxin domain II loop 1 as the binding site of Tenebrio molitor cadherin repeat CR12. *Insect Biochem Mol Biol*, v. 59, p. 50–57, 2015.

Apêndices

APÊNDICE A – ARQUIVOS GERADOS NOS EXPERIMENTOS - RESULTADOS BRUTOS

Os arquivos brutos gerados nos experimentos podem ser encontrados no diretório “experimentos/resultadosBrutos” da raiz do DVD de dados disponibilizado.

APÊNDICE B – ARQUIVOS GERADOS NOS EXPERIMENTOS - RESULTADOS FINAIS

Os arquivos finais gerados nos experimentos podem ser encontrados no diretório "experimentos/resultadosFinais" da raiz do DVD de dados disponibilizado e também *online* no endereço <https://github.com/davidbuzatto/CryProteinModelsComparisonData>.

APÊNDICE C – IMPLEMENTAÇÕES COMPUTACIONAIS

No decorrer do desenvolvimento desta pesquisa, foram realizadas diversas implementações de programas e trechos de código para a realização de tarefas, como o recorte dos dados dos resultados, bem como para o aprendizado de tecnologias e/ou bibliotecas específicas. A maioria das implementações foi feita utilizando a linguagem de programação Java e estão disponibilizadas como projetos da ferramenta de desenvolvimento NetBeans, versão 8.2. Sendo assim, na lista à seguir são descritas sucintamente tais implementações, sendo que as mesmas podem ser encontradas no diretório "implementações" da raiz do DVD de dados disponibilizado.

- **AlgoritmosEstruturaisGephi:** Geração dos dados para a criação do grafo de algoritmos de comparação estruturais de proteínas apresentado na Revisão da Literatura;
- **Alinhamentos:** Aplicação dos algoritmos de alinhamento local e alinhamento global em interface gráfica com o usuário com objetivo didático;
- **CryGetter:** Implementação da ferramenta "CryGetter". Pode também ser encontrada *online* no endereço <https://github.com/davidbuzatto/CryGetter> e sua versão executável para *download* no endereço <https://github.com/davidbuzatto/CryGetter>;
- **CryProteinModelsComparisonLab:** Implementação da ferramenta "*3-Domain Cry Protein Models Comparison Lab*". Pode também ser encontrada *online* no endereço <https://github.com/davidbuzatto/CryProteinModelsComparisonLab> e sua versão executável para navegadores de Internet no endereço <https://sbv.ifsp.edu.br/CryProteinModelsComparisonLab/>;
- **IndicesExperimento:** Obtenção dos índices dos recortes processados;
- **LabelsExperimentos:** Geração de *labels* para os recortes processados com o objetivo de preparar as imagens apresentadas nos Resultados;
- **ObtemRecortesSimilaridadeDali:** Processamento dos arquivos de saída gerados pelo Dali;
- **VerificadorModelos:** Verificador de integridade sequencial dos modelos do PDB;

- **ProcessaGeraDados:** *Pipeline* para a geração dos arquivos brutos e finais dos experimentos que utiliza vários trechos de código contidos nas implementações descritas acima.