Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Pós-Graduação em Ciência da Computação

SimVIZ - Um Ambiente Virtual de *Desktop* para Visualização e Análise de Múltiplas Trajetórias de Simulações de Proteínas

Ricardo Melo Czekster

Dissertação apresentada como requisito parcial à obtenção do grau de mestre em Ciência da Computação.

Orientador: Prof. Dr. Osmar Norberto de Souza

Porto Alegre, março de 2006.

À Rusalka.

Agradecimentos

Ao professor e orientador Osmar Norberto de Souza pelo incentivo, apoio, disponibilidade e dedicação durante o mestrado.

À minha família por me ensinar o valor da independência e do trabalho árduo para alcançar os objetivos. Aos meus pais, Waldemar Czekster e Elizabeth Melo Czekster e meus irmãos Gustavo Melo Czekster e Clarissa Melo Czekster por tudo.

Aos amigos e pesquisadores pelas opiniões durante a concepção deste trabalho. Em especial aos professores João Batista de Oliveira, Marcio Serolli Pinho e Milene Selbach Silveira bem como aos amigos Afonso Sales, André Benvenuti Trombetta, Edson Moreno, Ewerson Carvalho, Felipe Bacim de Araujo, Fausto Richetti Blanco, Luciano Copello Ost, Marco Aurélio Stelmar Netto, Mariana Lüderitz Kolberg, Paulo Fernandes, Patrick Ücker Calvetti, Pedro Velho, Rafael Guimaraes da Silva, Regis Augusto Poli Kopper e Thaïs Christina Webber dos Santos. Aos meus amigos Frederico Sório de Carvalho e Luiz Pedro Sório de Carvalho.

Aos colegas do Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas pela ajuda no entendimento dos conceitos de bioinformática estrutural, Ardala Breda, Cláudia Lemelle Fernandes e Evelyn Koeche Schroeder.

Gostaria de agradecer antecipadamente a todos que esqueci de agradecer acima...

Por fim, não menos importante, gostaria de agradecer aos números 4, 8, 15, 16, 23 e 42 por sua existência e pelo que representam. Obrigado.

À CAPES pelo auxílio financeiro.

“Labor Omni Vincit."

“Truth is ever to be found in the simplicity, and not in the multiplicity and confusion of things."
(Isaac Newton)

“The purpose of computing is insight, not numbers."
(Richard Hamming, 1962)

“You cannot teach a man anything, you can only help him to find it for himself."
(Galileu Galilei)

Resumo

Simulações de conformações de proteínas *in silico* geram quantidades massivas de dados que necessitam ser visualizados e analisados. Neste trabalho estamos aplicando conceitos e técnicas de Ambientes Virtuais de *Desktop* para auxiliar a análise de múltiplas trajetórias de simulações, procurando aumentar a experiência do usuário e desenvolvendo um ambiente de resolução de problemas para o processo de tomada de decisão. É apresentado o Ambiente *SimVIZ*, que integra a visualização à análise, melhorando o conhecimento prévio sobre as trajetórias. Este ambiente abre múltiplas janelas de desenho para múltiplas trajetórias, mostrando painéis informativos contendo saídas da simulação, informações associadas pelos usuários, Mapas de Contato, RMSD, Coordenadas Paralelas e o Gráfico de Ramachandran, todos em uma mesma cena.

Abstract

In silico protein conformation simulation generates massive amounts of data which needs to be properly visualized and analysed. We are applying Desktop Information-Rich Virtual Environments (*Desktop IRVE's*) techniques and concepts to aid multiple trajectory simulation analysis, seeking to augment user experience and developing a problem-solving environment to help the decision making process. We will present *SimVIZ*, an environment which integrates visualization to simulation analysis, improving previous knowledge about the trajectories. This environment opens multiple rendering windows for multiple trajectories showing informative panels with simulation outputs and user information, Contact Maps, RMSD plots, Parallel Coordinates and the Ramachandran Plot, all in a single scene.

Sumário

RESUMO v

ABSTRACT vi

LISTA DE TABELAS x

LISTA DE FIGURAS xi

LISTA DE SÍMBOLOS E ABREVIATURAS xv

Capítulo 1: Introdução 1

1.1 Organização 2

Capítulo 2: Bioinformática Estrutural e Simulação 3

2.1 Conceituação 3

2.2 Proteínas 4

2.2.1 Aminoácidos 4

2.3 Hierarquia Estrutural de Proteínas 9

2.4 Predição de Estruturas de Proteínas 11

2.5 Enovelamento ou Dobramento de Proteínas 13

2.6 Representações de Proteínas 14

2.7 Simulação de Proteínas 18

2.7.1 Ferramentas de Simulação 19

2.7.2 Entradas e Saídas para o *AMBER* 20

2.7.3 Análise de Simulações 20

Capítulo 3: Visualização e Ambientes Virtuais **24**

3.1 Visualização . . . 24

3.1.1 Visualização Multidimensional de Dados . . . 26

3.2 Ambientes Virtuais . . . 27

3.3 Ambientes Virtuais Ricos em Informação . . . 28

3.3.1 Definição formal de *IRVE's* . . . 28

3.3.2 Especificações para o Tratamento das Informações . . . 29

3.3.3 Classificação de Técnicas de Leiaute de Textos . . . 31

3.3.4 Objetos Semânticos . . . 33

3.4 Ambientes Virtuais de *Desktop* Ricos em Informação . . . 33

Capítulo 4: Trabalhos Relacionados **35**

4.1 Sistemas de Visualização de Proteínas . . . 35

4.1.1 *PyMOL Molecular Graphics System* . . . 35

4.1.2 *DeepView - Swiss PDB Viewer* . . . 36

4.1.3 *VMD - Visual Molecular Dynamics* . . . 38

4.2 Sistemas de Análise de Trajetórias de Simulações . . . 38

4.2.1 *ptraj/rdparm* . . . 39

4.2.2 Outros *Softwares* e Bibliotecas . . . 40

4.3 *Virtual Data Visualizer* . . . 41

4.4 *PathSim Visualizer* . . . 41

4.5 Discussão dos Trabalhos Relacionados . . . 42

Capítulo 5: O Ambiente *SimVIZ* **43**

5.1 Descrição do Ambiente *SimVIZ* . . . 43

5.2 Módulos do Ambiente *SimVIZ* . . . 46

5.2.1 Entrada de Dados, Interface, Controles de Animação e Montador de Topologia . . . 47

5.2.2 Enriquecimento de Informações . . . 48

5.2.3 Mapeamentos . . . 49

5.2.4 Gerenciador de Informações . . . 49

5.2.5 *Rendering* . . . 55

5.3 *SimVIZ* - Ambiente Virtual de *Desktop* Rico em Informação 59
5.3.1 Representação Textual de Átomos e Aminoácidos na Cena de Desenho . . 60
5.3.2 Localização e Associação de Informações 64
5.3.3 Painéis Informativos 66
5.3.4 Integração dos Elementos Gráficos na Cena de Desenho 68
5.4 Arquitetura do Ambiente *SimVIZ* 71
5.5 *Feedback* de Usuários 77
5.6 Considerações Finais Sobre o Ambiente *SimVIZ* 78

Capítulo 6: Conclusões **80**
6.1 Trabalhos Futuros 81

REFERÊNCIAS **84**

Apêndice A: Manual de Usuário do Ambiente *SimVIZ* **89**
A.1 Pré-requisitos 89
A.2 Detalhes de Implementação 89
A.3 Funcionalidades 90
A.4 Interface Gráfica de Usuário 91

Lista de Tabelas

2.1 Listagem dos 20 aminoácidos 5
2.2 Código de 1 letra das estruturas secundárias de proteínas 11

Lista de Figuras

2.1 Ângulos phi (ϕ) e psi (ψ) de uma proteína [3]. 5
2.2 Gráfico de Ramachandran [9]. 6
2.3 Aminoácidos hidrofóbico-alifáticos. (a) alanina, (b) valina, (c) leucina, (d) isoleucina. 7
2.4 Aminoácidos hidrofóbico-aromáticos. (a) fenilalanina, (b) tirosina, (c) triptofano. 7
2.5 Aminoácidos neutro-polares. (a) serina, (b) treonina, (c) cisteína, (d) metionina, (e) asparagina, (f) glutamina. 8
2.6 Aminoácidos ácidos. (a) ácido aspártico, (b) ácido glutâmico. 9
2.7 Aminoácidos básicos. (a) histidina, (b) lisina, (c) arginina. 9
2.8 Aminoácidos com conformações importantes. (a) glicina, (b) prolina. 10
2.9 Classificação de proteínas por estrutura: (a) Estrutura primária (listagem dos aminoácidos). Estruturas secundárias em (b) α-hélice (c) voltas e alças (d) fita-β. (e) Estrutura terciária [5]. 11
2.10 Estrutura quaternária da proteína hemoglobina [25]. 12
2.11 Processo de enovelamento de proteínas. 14
2.12 Proteína com código *PDB 1A00* representada pelo formato *Lines* [31]. 15
2.13 Representação com o formato *Bonds* da proteína com código *PDB 1A00* [31]. . . 15
2.14 Representação da proteína com código *PDB 1A00* com o formato *Cartoon* [31]. . 16
2.15 Proteína com código *PDB 1A00* desenhada com o formato *VDW* [31]. 17
2.16 Representação da proteína com código *PDB 1A00* na forma *CPK* [31]. 17
2.17 Representação da proteína com código *PDB 1A00* com a forma *New_ ribbons* [31]. 18
2.18 Gráfico do *RMSD* (medido em ångströms) de uma trajetória em função do tempo de simulação (medido em picossegundos) para o arquivo "rms_ca.dat" (gerado pelo *ptraj*). 21

2.19 Mapa de Contatos da proteína com código *PDB 1ZDB*. 23

3.1 *Pipeline* para visualização de informações [45]. 25
3.2 Visualização multidimensional com Coordenadas Paralelas [32]. 27
3.3 Taxonomia de técnicas de leiaute de textos para *IRVE's* [20]. 32

4.1 Interface gráfica do sistema *PyMOL* da proteína com código *PDB 1A00*. 36
4.2 Proteína com código *PDB 1A00* representada no sistema *DeepView*. 37
4.3 Interface gráfica do *VMD* da proteína com o código *PDB 1A00*. Em a) Janela de visualização. b) Janela auxiliar de representações gráficas e c) Janela principal do sistema. 39

5.1 Fluxograma inicial de execução do Ambiente *SimVIZ*. 45
5.2 Proteína da Estrutura de Referência com código *PDB* 1ZDB. Em (a) hélice 1 e (b) hélice 2. 46
5.3 Conjunto de Módulos do Ambiente *SimVIZ*. 47
5.4 Gráfico de Ramachandran. (a) Proteína no passo de simulação. (b) Estrutura de Referência. 50
5.5 Mapa de Contatos. (a) Proteína no passo de simulação. (b) Estrutura de Referência. 51
5.6 Gráfico do *RMSD* de quatro diferentes arquivos de análise. 52
5.7 Visualização multidimensional de trajetória com Coordenadas Paralelas no décimo nanossegundo do tempo de simulação. 52
5.8 Visualização multidimensional de trajetória com Coordenadas Paralelas no último tempo de simulação (centésimo). 53
5.9 Gráficos bidimensionais relativos às saídas de simulação. 53
5.10 Gráficos bidimensionais com as informações complementares. 54
5.11 Contagens de átomos e aminoácidos. Em a) contagens por elemento químico e em b) contagens por aminoácidos (utilizando o código de uma letra). 54
5.12 Painéis informativos contendo (a) dados de saída do *AMBER* (b) listagem dos aminoácidos e estrutura secundária e (c) parâmetros geométricos da proteína. . . 55
5.13 Representação no formato *Lines* de uma proteína. 56
5.14 Representação no formato *Bonds* de uma proteína colorida de acordo com o nome do aminoácido. 57

5.15 Representação no formato *CPK* de uma proteína, colorida pelo *backbone* (a cor verde mostra os átomos do *backbone* e a cor vermelha exibe as cadeias laterais). . 57
5.16 Representação no formato *VDW* de uma proteína colorida pelo nome do elemento químico. 58
5.17 *Backbone* de uma proteína representada pelo formato *CPK* e colorido pelo nome do elemento químico, mostrando apenas os átomos de carbono (em azul claro) e nitrogênio (em azul escuro). 58
5.18 Janela de visualização do Ambiente *SimVIZ* ilustrando os tempos de 1 a 9 nanossegundos da trajetória de simulação. 60
5.19 Detalhe das estruturas secundárias nos tempos de 1 a 9 nanossegundos da trajetória de simulação. 61
5.20 Detalhe das saídas de simulação nos tempos de 1 a 9 nanossegundos da trajetória de simulação. 62
5.21 Representação textual de todos os aminoácidos de uma proteína. 63
5.22 Representação textual do nome do aminoácido *Alanina* utilizando o seu código de três letras. 63
5.23 Representação textual de todos os átomos de uma proteína sem escala. 64
5.24 Representação textual do nome dos átomos de carbono do *backbone* de uma proteína. 65
5.25 Representação textual do nome de todos os átomos do *backbone* de uma proteína com escala. 65
5.26 Diagrama esquemático dos painéis informativos e regiões informativas do Ambiente *SimVIZ*. 67
5.27 Representação de proteína e os painéis informativos. 69
5.28 Painéis informativos e gráficos bidimensionais. 70
5.29 Painéis informativos e contagens de átomos e aminoácidos. 71
5.30 Representação do números de átomos e do Gráfico de Ramachandran. 72
5.31 Mapa de Contatos da proteína e da Estrutura de Referência. 73
5.32 Representação de proteína e visualização das Coordenadas Paralelas. 74
5.33 Representação de proteína e visualização do gráfico de *RMSD* para 4 arquivos de análise. 75
5.34 Visualização de informações associadas (a) globais, (b) ao passo de simulação e (c) a dois aminoácidos específicos. 76

5.35 Arquitetura do Ambiente *SimVIZ* utilizando a notação simplificada do Diagrama de Classes de UML. 76

A.1 Interface geral do Ambiente *SimVIZ*. 91
A.2 Menu *File* do Ambiente *SimVIZ*. 92
A.3 Menu *View* do Ambiente *SimVIZ*. 92
A.4 Janela de visualização do Ambiente *SimVIZ*. 93
A.5 Menu *File* > *Open Simulation* do Ambiente *SimVIZ*. 93
A.6 Menu *View* > *Representation* do Ambiente *SimVIZ*. 94
A.7 Menu *View* > *Information Manager* do Ambiente *SimVIZ*. 95

Lista de Símbolos e Abreviaturas

PDB	*Protein Data Bank*	4
VDW	*van der Waals*	14
CPK	*Corey, Pauling, Koltun*	16
DM	*Dinâmica Molecular*	18
MD	*Molecular Dynamics*	18
AMBER	*Assisted Model Building with Energy Refinement*	19
CHARMM	*Chemistry at Harvard Macromolecular Mechanics*	19
GROMOS	*Groningen Molecular Simulation*	19
RMSD	*Root Mean-Square Deviation*	21
VE	*Virtual Environment*	27
IRVE	*Information-Rich Virtual Environment*	28
AR	*Augmented Reality*	28
HUD	*Heads-up Display*	32
GUI	*Graphical User Interface*	46
UML	*Unified Modelling Language*	71

Capítulo 1

Introdução

Simulações feitas por computadores, ou *in silico*, como são conhecidas, baseiam-se no estudo de problemas reais através do mapeamento para modelos abstratos que os representem. Um caso especial é a simulação pelo método da Dinâmica Molecular (DM) de estruturas tridimensionais de proteínas. Tais simulações geram extensos dados de saída que precisam ser analisados. A procura por informações relevantes não é uma tarefa trivial e os esforços são direcionados para o desenvolvimento de ferramentas interativas que permitam destacar regiões importantes dos dados [51, 40, 24, 28].

Cada simulação pelo método da DM gera, a partir de um conjunto de parâmetros iniciais, uma trajetória (um histórico) das estruturas tridimensionais assumidas por proteínas, retornando informações sobre o estado do sistema, tais como a energia, a temperatura, a pressão e as coordenadas atômicas. É comum produzir inúmeras simulações a partir da permutação de diferentes parâmetros de entrada, desta forma aumentando o número de análises a serem realizadas.

O procedimento usual na fase de análise é utilizar ferramentas específicas para a construção de gráficos (normalmente bidimensionais) que mostram o comportamento das saídas ao longo da simulação. As ferramentas atuais de visualização e análise permitem que sejam exibidas múltiplas simulações mas, devido ao fato de serem dotadas de objetivos exclusivos (unicamente de visualização ou unicamente de análise), são normalmente não escaláveis e com a arquitetura não documentada. Este fato impede que sejam estensíveis para problemas mais específicos.

O objetivo deste trabalho é propor e desenvolver um ambiente integrado de visualização e análise de múltiplas trajetórias de simulação de proteínas pelo método da Dinâmica Molecular. Para tanto, serão utilizados conceitos de Visualização Científica, Visualização de Informações e Ambientes Virtuais de *Desktop* Ricos em Informação [20] (maiores detalhes na Seção 3.3).

Este ambiente, denominado *SimVIZ*, objetiva auxiliar no processo de visualização e análise dos dados produzidos por simulações pelo método da Dinâmica Molecular, destacando e apresentando informações relevantes para a tomada de decisões.

1.1 Organização

Este trabalho está organizado da seguinte forma: no Capítulo 2 são definidos os aspectos teóricos sobre proteínas, trajetórias e simulações. O Capítulo 3 trata sobre ambientes virtuais e visualização, seguido dos trabalhos relacionados no Capítulo 4. O Capítulo 5 apresenta o Ambiente *SimVIZ*. Por fim, o Capítulo 6 descreve as conclusões e os trabalhos futuros.

Capítulo 2

Bioinformática Estrutural e Simulação

A Bioinformática Estrutural é uma área multidisplinar do conhecimento que une conceitos de áreas como Matemática, Estatística, Física, Química e Ciência da Computação. Este capítulo introduz esta área de estudo e trata sobre proteínas, aminoácidos e simulações utilizando o método da Dinâmica Molecular.

2.1 Conceituação

Bioinformática Estrutural, ou apenas bioinformática, trata da organização, recuperação, representação e manipulação de dados biológicos, aplicando técnicas de disciplinas como Matemática, Estatística, Física, Química e Ciência da Computação [39]. A bioinformática preocupa-se com as informações associadas a moléculas biológicas, agregando-as em bancos de dados de diferentes origens. A maior parte das análises realizadas referem-se a três fontes de dados principais: seqüências de DNA e proteínas, estruturas macromoleculares e resultados de experimentos genômicos [39].

Sequências de DNA são sequências de *strings* com quatro tipos de caracteres chamados de bases (adenina, citosina, guanina e timina, ou A, C, G e T, respectivamente). O repositório *GenBank* [7] armazena sequências deste tipo e o seu tamanho é de 100 bilhões de bases de 165.000 organismos, em Outubro de 2005 [16]. Ainda neste mesmo banco de dados, há as seqüências de proteínas que são formadas por conjuntos de aminoácidos (também chamados de "resíduos").

Dados de Outubro de 2005 confirmavam a existência de 195.058 seqüências no repositório do *Swiss-Prot Knowledgebase* [14], que é um banco de dados central, consistente e confiável de proteínas e anotações (textos explicativos contendo a função da proteína, as estruturas secondárias e

similaridade com outras proteínas). Um dos objetivos do *Swiss-Prot* é manter uma redundância mínima entre as seqüências, ou seja, procura descobrir seqüências similares e agregá-las em uma mesma entrada no banco de dados [18, 14, 6].

Estruturas macromoleculares representam uma forma mais complexa de informação. Dados de Outubro de 2005 mostram que já foram depositadas 4.300 estruturas únicas resolvidas no repositório de arquivos PDB (*Protein Data Bank* [17]), mas por volta de 33 mil estruturas foram obtidas experimentalmente (através de Cristalografia por Difração de Raios X ou Ressonância Magnética Nuclear) ou através de modelos teóricos [10, 39].

Por fim, os resultados de experimentos genômicos são sequências completas do genoma, guardadas em respositórios como o *GenBank* (E.U.A.) [16, 7], *EMBL* (Europa) [36, 13] e *DDBJ* (Japão) [41, 4].

2.2 Proteínas

Proteínas são macromoléculas constituídas por seqüências de aminoácidos e desempenham funções específicas dentro da célula. A estrutura tridimensional de uma proteína determina sua função e surge da interação entre os seus aminoácidos constituintes. O processo pelo qual esta seqüência atinge sua conformação correta, atingindo o seu estado biológico ativo (onde realiza função) é chamado de enovelamento ou dobramento (ou, em inglês, *folding*) [21].

O enovelamento de uma proteína gera uma estrutura tridimensional particular a partir de uma estrutura linear, determinada por sua seqüência de aminoácidos e o ambiente a sua volta. Este é o problema central da biologia molecular estrutural: predizer a estrutura tridimensional de uma proteína a partir unicamente dos seus aminoácidos [21].

2.2.1 Aminoácidos

Todas as proteínas são construídas a partir de moléculas menores, chamadas aminoácidos. A Tabela 2.1 mostra os 20 aminoácidos naturais, cujas propriedades químicas são conhecidas com precisão. Cada aminoácido compartilha uma estrutura básica, formada por um átomo de carbono central (C), um grupo amino (NH_3^+), um grupo carboxila ($COOH^-$) e uma cadeia lateral (R). Estes grupos químicos determinam o funcionamento da molécula [21].

Cadeias de aminoácidos são montadas por uma reação que ocorre entre o átomo de nitrogênio do grupo amino de um aminoácido e o átomo de carbono do grupo carboxila de outro, ligando os

Tabela 2.1: Listagem dos 20 aminoácidos

Código de 1 letra	Código de 3 letras	Nome do aminoácido
A	ALA	Alanina
R	ARG	Arginina
N	ASN	Asparagina
D	ASP	Ácido Aspártico
E	GLU	Ácido Glutâmico
C	CYS	Cisteína
F	PHE	Fenilalanina
G	GLY	Glicina
Q	GLN	Glutamina
H	HIS	Histidina
I	ILE	Isoleucina
L	LEU	Leucina
K	LYS	Lisina
M	MET	Metionina
P	PRO	Prolina
S	SER	Serina
Y	TYR	Tirosina
T	THR	Treonina
W	TRP	Triptofano
V	VAL	Valina

dois aminoácidos e liberando uma molécula de água. A ligação é chamada de ligação peptídica.

As partes variantes dos aminoácidos são chamadas de cadeias laterais; os dois carbonos e o nitrogênio no núcleo são denominados de *backbone*. Ligações peptídicas são conectadas entre si pelos *backbones* de seqüências de aminoácidos. Esta ligação pode ser caracterizada como tendo dois graus de liberdade rotacional: ângulos phi (ϕ) e psi (ψ) (conforme Figura 2.1) [21].

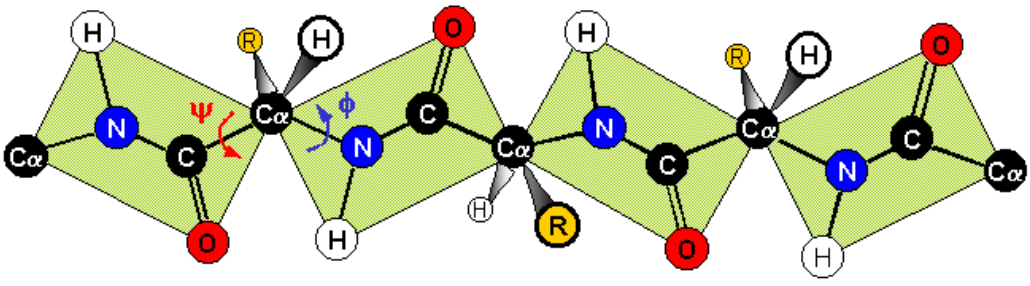

Figura 2.1: Ângulos phi (ϕ) e psi (ψ) de uma proteína [3].

Muitas combinações de ângulos ϕ e ψ não são possíveis, devido às colisões entre as cadeias

laterais e a cadeia principal. Estes pares de ângulos são normalmente representados com um gráfico chamado *Gráfico de Ramachandran*. Este gráfico informa as combinações de ângulos permitidas às unidades peptídicas [21].

A estrutura é examinada para verificar contatos entre os átomos, seguindo as dimensões correspondentes do seu raio de van der Waals. A Figura 2.2 mostra o Gráfico de Ramachandran. As regiões delimitadas pela cor verde correspondem a regiões permitidas, ou seja, onde não existem colisões entre os átomos, e podem ser divididas em áreas que formam α-hélices ou fitas-β (essas estruturas são explicadas a seguir, na Seção 2.3). Para ϕ maior que zero, indica a formação de α-hélices levógiras e para ϕ menor que zero, α-hélices dextrógiras. Já as regiões delimitadas pela borda de cor verde e azul mostram combinações de ângulos ϕ e ψ onde o raio de van der Waals usado no cálculo foi levemente modificado para um valor menor, permitindo áreas adicionais sem a ocorrência de colisões. As demais regiões são proibidas [9].

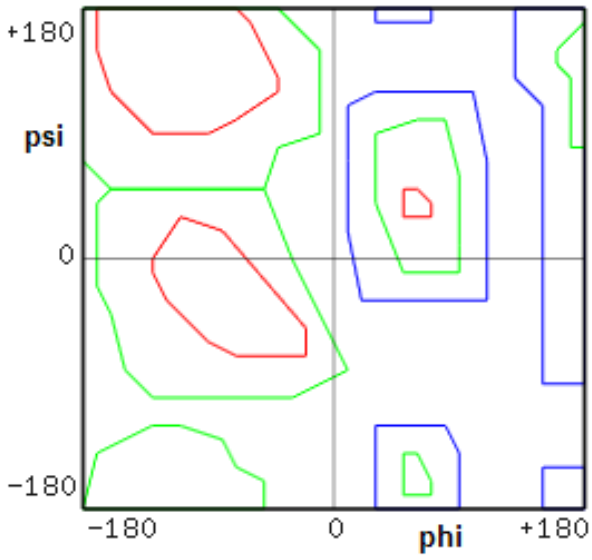

Figura 2.2: Gráfico de Ramachandran [9].

Estas cadeias podem ser classificadas de acordo com suas propriedades físico-químicas, como os aminoácidos hidrofóbico-alifáticos, hidrofóbico-aromáticos, neutro-polares, ácidos, básicos e de conformações importantes [21].

Os aminoácidos hidrofóbico-alifáticos são compostos por quatro aminoácidos: alanina, valina, leucina e isoleucina, normalmente encontrados no interior de proteínas (pela sua natureza hidrofóbica), conforme Figura 2.3 [21].

A categoria dos aminoácidos hidrofóbico-aromáticos é formada por fenilalanina, tirosina e

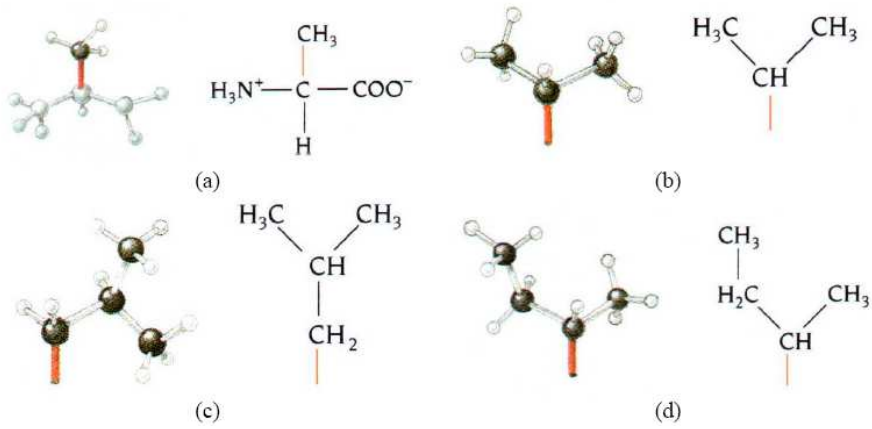

Figura 2.3: Aminoácidos hidrofóbico-alifáticos. (a) alanina, (b) valina, (c) leucina, (d) isoleucina.

triptofano. Estes aminoácidos também podem ser encontrados predominantemente no interior de proteínas. A Figura 2.4 demonstra a sua estrutura e composição [21].

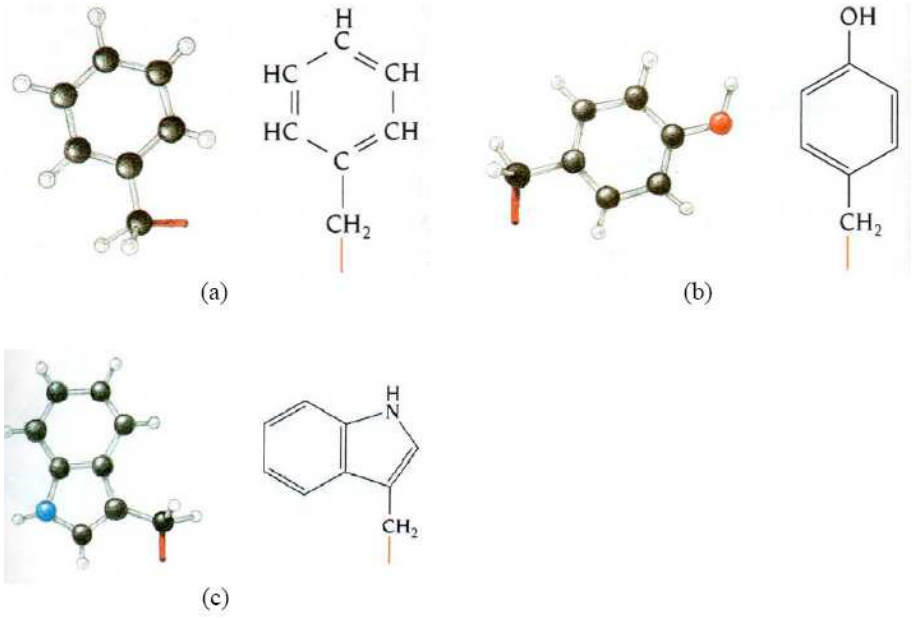

Figura 2.4: Aminoácidos hidrofóbico-aromáticos. (a) fenilalanina, (b) tirosina, (c) triptofano.

Outra categoria é a dos aminoácidos neutro-polares. Seis aminoácidos formam este grupo (serina, treonina, cisteína, metionina, asparagina e glutamina), caracterizado por possuir grupos polares que não ionizam facilmente. Serina e treonina possuem grupos hidroxila nas suas cadeias laterais, próximas à cadeia principal formando ligações de hidrogênio entre si, influenciando a conformação do polipeptídeo (de acordo com a Figura 2.5) [21].

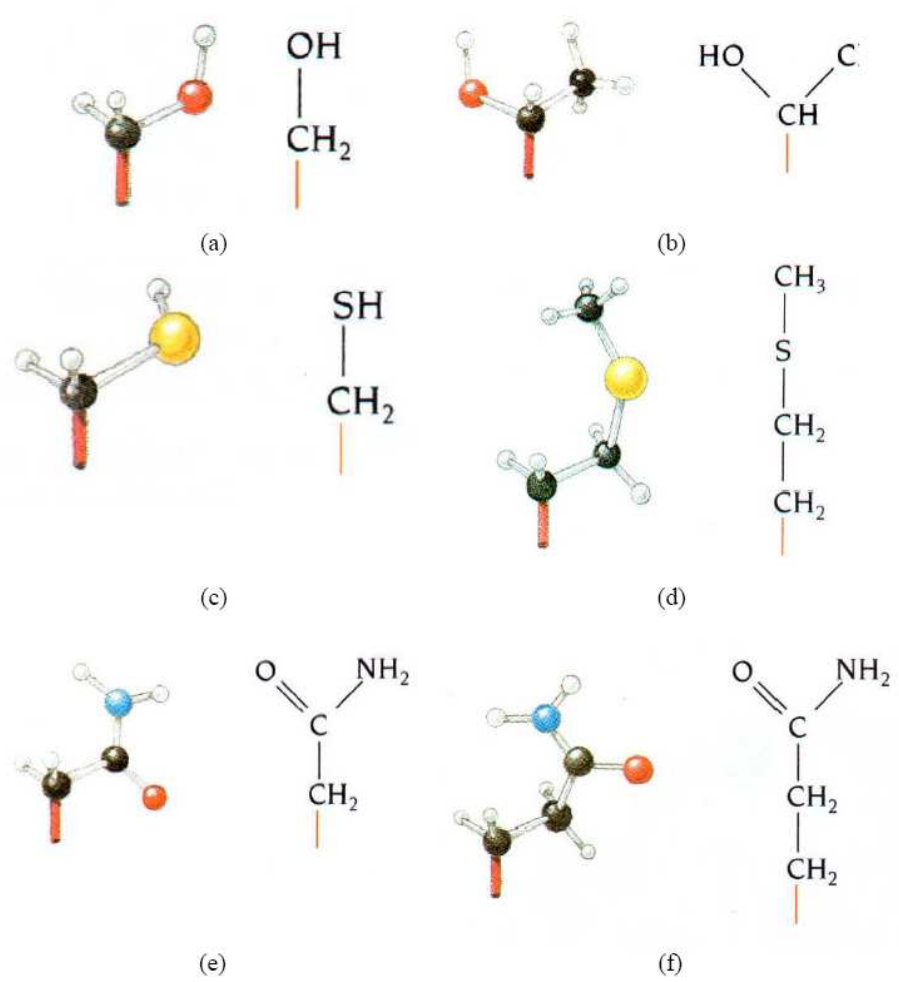

Figura 2.5: Aminoácidos neutro-polares. (a) serina, (b) treonina, (c) cisteína, (d) metionina, (e) asparagina, (f) glutamina.

Os aminoácidos ácidos são formados por ácido aspártico e ácido glutâmico (Figura 2.6). A natureza polar destes resíduos faz com que sejam encontrados na superfície de proteínas, pois interagem favoravelmente com moléculas solventes [21].

O grupo dos aminoácidos básicos é formado por histidina, lisina e arginina. Estes aminoácidos ocorrem freqüentemente em sítios ativos (a Figura 2.7 mostra os aminoácidos desta classe) [21].

Por fim, existem dois aminoácidos com conformações simples, porém importantes, que influenciam de forma direta a estrutura de uma proteína. Este grupo é formado pela glicina e pela prolina (Figura 2.8) [21].

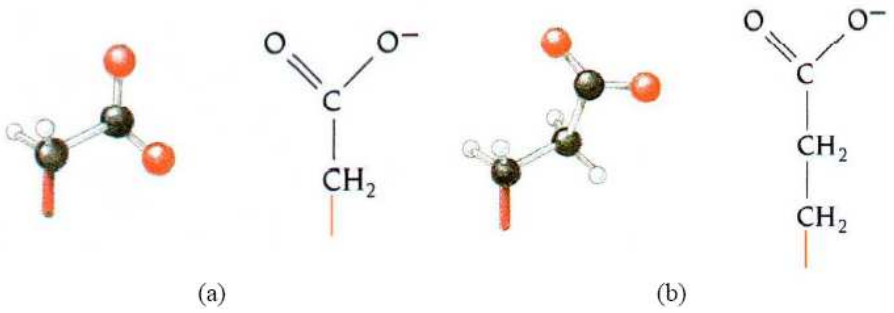

Figura 2.6: Aminoácidos ácidos. (a) ácido aspártico, (b) ácido glutâmico.

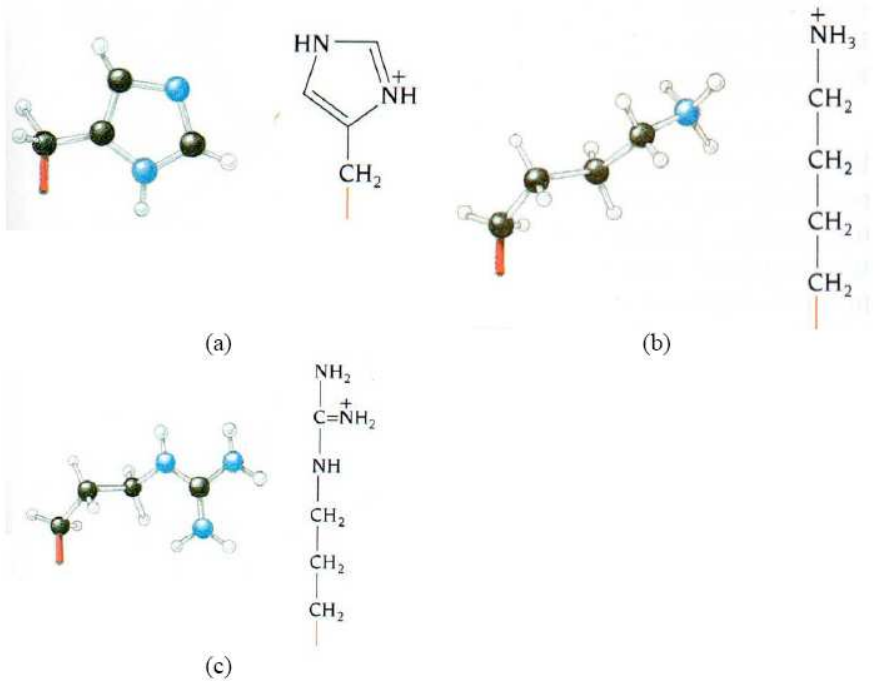

Figura 2.7: Aminoácidos básicos. (a) histidina, (b) lisina, (c) arginina.

2.3 Hierarquia Estrutural de Proteínas

As proteínas são classificadas nas seguintes hierarquias estruturais [21]:

- Estrutura Primária: trata-se do conjunto de aminoácidos que formam uma proteína. Cada aminoácido da seqüência associa-se com outros aminoácidos conservando o máximo de energia. O início da seqüência é caracterizado pelo grupo N (NH_3^+) e no fim, pelo grupo terminal carboxila (COO^-);

- Estrutura Secundária: de acordo com os aminoácidos da seqüência primária, uma proteína

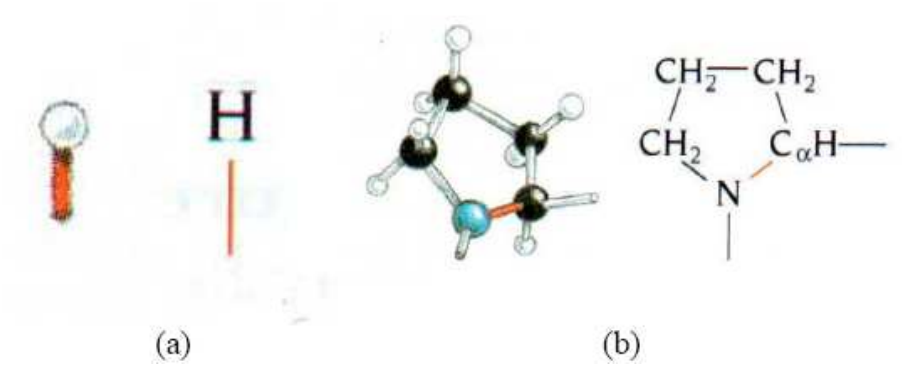

Figura 2.8: Aminoácidos com conformações importantes. (a) glicina, (b) prolina.

forma estruturas secundárias, tais como α-hélices, fitas-β, voltas e alças. Estas formações são favorecidas pelas ligações de hidrogênio entre os aminoácidos existentes;

- Estrutura Terciária: as estruturas terciárias são composições no espaço tridimensional (3D) dos elementos estruturais secundários, como mostra a Figura 2.9;

- Estrutura Quaternária: a proteína final pode conter muitas cadeias polipeptídicas arranjadas em uma estrutura quaternária. São associações de proteínas já organizadas em nível terciário. Estas estruturas podem unirem-se formando um sítio ativo. A Figura 2.10 mostra a proteína hemoglobina (seu código no repositório do PDB é *1A00*), onde cada cor representa uma subunidade de estrutura terciária (a Seção 2.6 explica as formas de representações protéicas).

Para o reconhecimento de elementos estruturais secundários em coordenadas atômicas de proteínas são necessárias implementações de algoritmos específicos, como *STRIDE* [27], *FSSP* [30] ou *DSSP* [34]. Para o escopo deste trabalho, estudaremos, em maiores detalhes, o *STRIDE*. Trata-se de um *software* para o reconhecimento de elementos estruturais secundários em proteínas que recebe como entrada um arquivo do formato *PDB*. Esta biblioteca de funções realiza as mesmas tarefas que o *DSSP*, mas o seu diferencial é utilizar tanto informações de energia de ligação dos hidrogênios, quanto os ângulos diedros da cadeia principal. O *STRIDE* executa na plataforma *Windows* e *Linux*, e possui o código-fonte aberto, podendo ser adicionado (*linked*) em um projeto. O *software* reconhece as estruturas secundárias do tipo α-hélice, fita-β, volta e alça.

O *STRIDE* calcula as estruturas secundárias e define os seus códigos de 1 letra, conforme mostra a Tabela 2.2 [27].

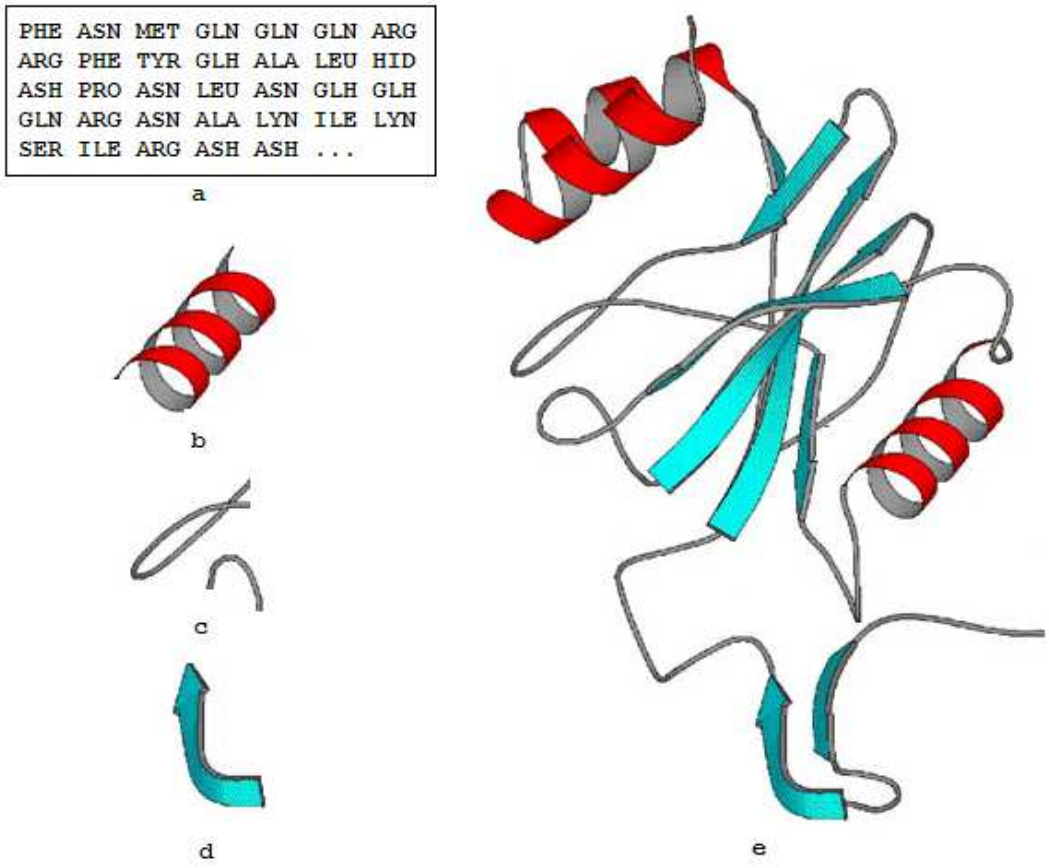

Figura 2.9: Classificação de proteínas por estrutura: (a) Estrutura primária (listagem dos aminoácidos). Estruturas secundárias em (b) α-hélice (c) voltas e alças (d) fita-β. (e) Estrutura terciária [5].

Tabela 2.2: Código de 1 letra das estruturas secundárias de proteínas

Código de 1 letra	**Estrutura Secundária**
C	Alça (*Coil*)
H	α-hélice (α-*helix*)
G	Hélice $3-10$ ($3-10$-*helix*)
I	Hélice-π (π-*helix*)
E	Folha-β (β-*sheet*)
B	*Isolated bridge*
T	Volta (*Turn*)

2.4 Predição de Estruturas de Proteínas

Conforme mencionamos no início deste capítulo, a identificação de seqüências de aminoácidos para estruturas tridimensionais de proteínas é um problema recorrente em bioinformática. A

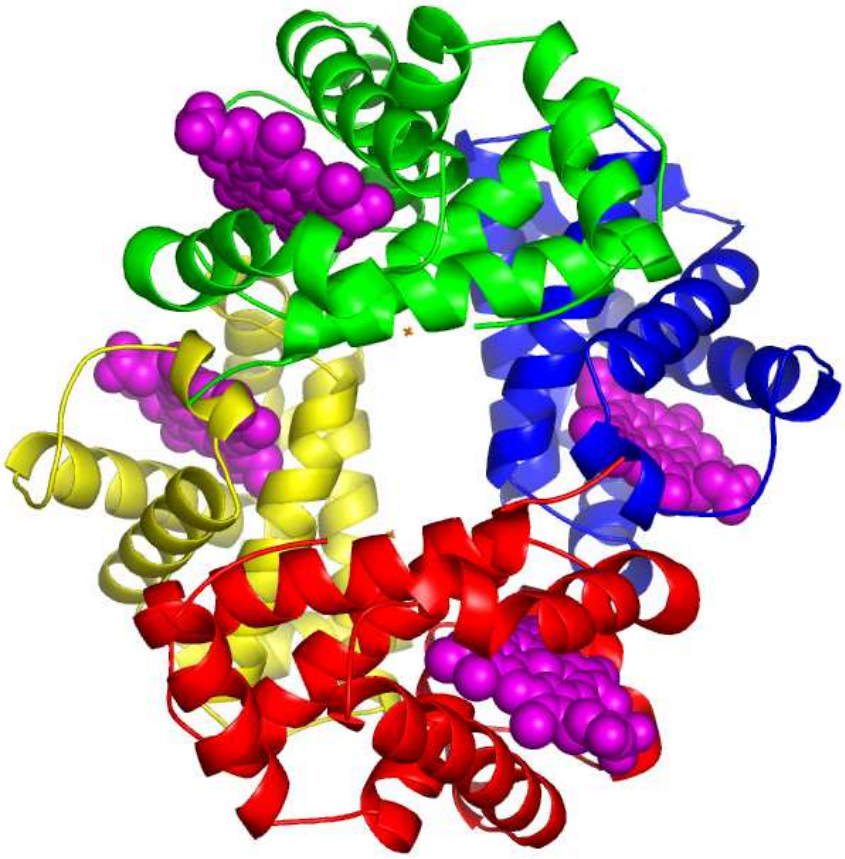

Figura 2.10: Estrutura quaternária da proteína hemoglobina [25].

obtenção da estrutura terciária final exata, unicamente a partir da estrutura primária, é computacionalmente custosa e implica no desenvolvimento de métodos confiáveis para a predição das estruturas [21].

Existem três métodos usados para prever a estrutura terciária de uma proteína: comparação por homologia, *threading* e *ab initio* (primeiros princípios):

- Homologia: a modelagem por homologia é baseada no conceito de que novas proteínas evoluem gradualmente a partir de outras pré-existentes através de substituições, adições ou remoções de aminoácidos. Muitas proteínas compartilham funções e estruturas e normalmente existe similaridade entre seqüências. A qualidade da modelagem depende se existe ou não uma ou mais estruturas nos bancos de dados existentes [21];

- *Threading*: *threading* é o método pelo qual realiza-se uma pesquisa em uma biblioteca de estruturas únicas e representativas em busca de estruturas análogas à seqüência alvo e fundamenta-se na teoria de que existe um número limitado de conformações estruturais de

proteínas. O método possui alto custo computacional, já que cada entrada na biblioteca precisa ser alinhada de todas as formas possíveis. O principal requisito para *threading* é acessar um banco de dados contendo os diferentes enovelamentos de proteínas disponíveis [21];

- *Ab initio* ou primeiros princípios: na falta de estruturas conhecidas para serem usadas como base, métodos *ab initio* são usados para prever a estrutura unicamente a partir da informação sobre as seqüências. Neste método é considerada apenas a seqüência alvo e as propriedades dos aminoácidos. Este método gera conformações de estruturas terciárias e para cada uma avalia a energia através de funções potenciais. Esta avaliação é baseada na Hipótese Termodinâmica que estabelece que a estrutura nativa da proteína é aquela onde a energia livre alcança o seu valor mínimo global [21, 50].

2.5 Enovelamento ou Dobramento de Proteínas

O enovelamento é o processo pelo qual a seqüência de aminoácidos de uma proteína atinge sua conformação tridimensional termodinamicamente mais estável, realizando função dentro da célula. Entender este processo é crucial para regular atividades biológicas. O número total de possibilidades conformacionais que uma proteína possui é extremamente elevado. Uma busca sistemática pela estrutura correta acarretaria uma quantidade de tempo elevada para ser concluída. Entretanto, o enovelamento realiza uma busca estocástica no espaço conformacional, ou seja, não percorre todas as possibilidades [26].

A Figura 2.11 mostra uma simulação por DM de um enovelamento: à medida que o tempo passa (de *zero* a 2000 picossegundos), a proteína começa na forma estendida e varia sua posição à procura de uma estrutura mais energeticamente estável. Para saber se a trajetória está convergindo para uma estrutura alvo, utiliza-se a chamada *Estrutura de Referência*. O último desenho da Figura 2.11 (sob a legenda "Referência") mostra a estrutura de referência que deve ser atingida por uma determinada simulação. Um outro aspecto interessante sobre esta figura é o fato de que as estruturas secundárias também variam em função do tempo, ou seja, são novamente preditas a cada passo de simulação e influem na representação da proteína.

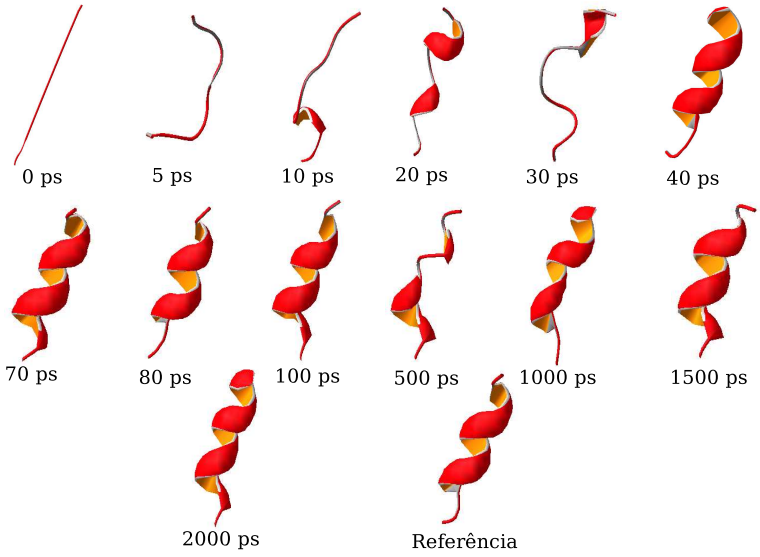

Figura 2.11: Processo de enovelamento de proteínas.

2.6 Representações de Proteínas

Os modelos reais de moléculas (coleções de átomos que são tanto partículas quanto ondas [21]) são difíceis de serem visualizados, justificando o uso de modelos mais simplificados, como *ball-and-stick* que representam os átomos como esferas e as ligações como cilindros. Outras representações ilustram propriedades da superfície das moléculas. Entretanto, muitas vezes são necessárias formas mais abstratas de representação, que informem as estruturas secundárias de proteínas, como α-hélices e folhas-β, por exemplo [21].

Proteínas são visualizadas de diferentes maneiras como, por exemplo, *Lines*, *Bonds*, *VDW*, *Ribbons*, *Cartoon* e *New_ribbons*. A justificativa para múltiplas formas de visualização de proteínas relaciona-se com a análise que está sendo conduzida. Determinadas características ficarão mais ou menos evidentes dependendo da representação utilizada. A seguir, uma descrição mais detalhada de cada tipo:

- *Lines*: é a forma mais simples e rápida para visualização molecular. Esta representação desenha linhas nas ligações entre os átomos das moléculas de uma proteína, sem desenhar os átomos propriamente ditos. A forma de colorir foi relativa a cada elemento químico (utiliza uma cor para cada *Carbono*, *Hidrogênio*, *Oxigênio*, *Nitrogênio*, *Fósforo* ou *Enxofre*). A Figura 2.12 mostra uma proteína construída com *Lines*;

- *Bonds*: é uma representação semelhante à *Lines*, mas representa as ligações com cilindros,

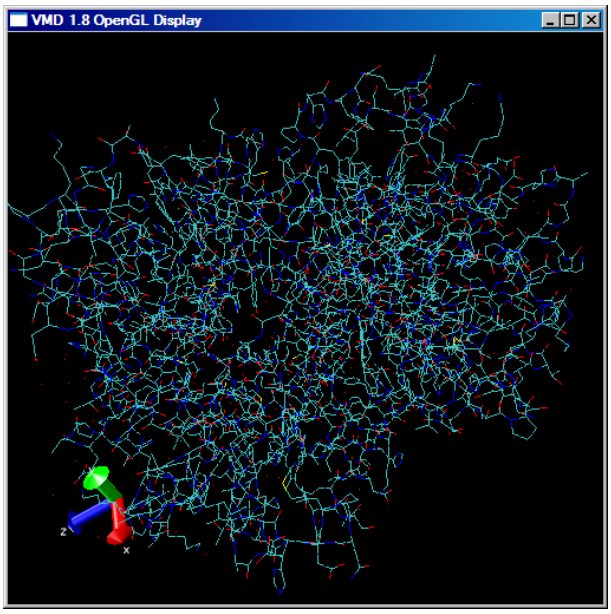

Figura 2.12: Proteína com código *PDB 1A00* representada pelo formato *Lines* [31].

ao invés de linhas, conforme ilustrado pela Figura 2.13 (colorindo de acordo com o elemento químico);

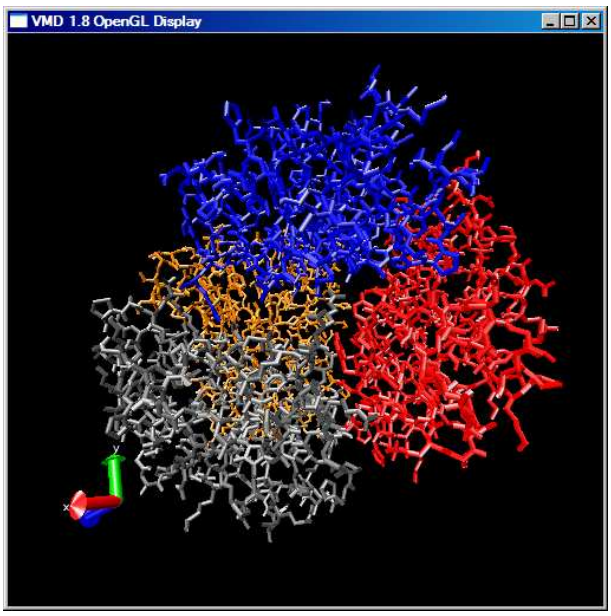

Figura 2.13: Representação com o formato *Bonds* da proteína com código *PDB 1A00* [31].

- *Cartoon*: mostra as α-hélices como cilindros maiores e o resto das estruturas secundárias como cilindros menores. A Figura 2.14 mostra uma representação deste tipo (nesta re-

presentação, escolhemos colorir pelo identificador da cadeia, ou seja, diferentes cores para cada estrutura terciária contida no arquivo *PDB*);

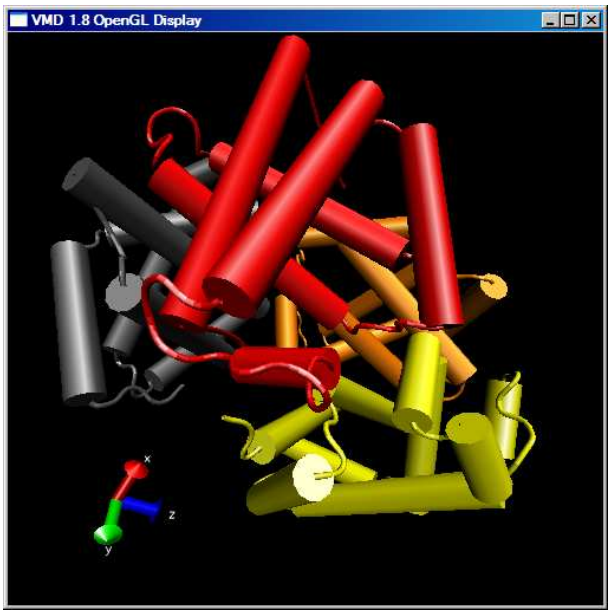

Figura 2.14: Representação da proteína com código *PDB 1A00* com o formato *Cartoon* [31].

- *VDW*: esta representação utiliza o raio de van der Waals para desenhar a estrutura. Este raio varia para cada átomo da molécula e esta visualização é útil para inferir o volume de uma proteína. A Figura 2.15 mostra a representação *VDW*, colorindo conforme o elemento químico;

- *CPK*: "*CPK*" é uma combinação de *Bonds* e *VDW*, desenhando os átomos como esferas e as ligações como cilindros, de acordo com a Figura 2.16. A forma de colorir usada foi pelo identificador da cadeia;

- *Ribbons*: *Ribbons* desenha cada estrutura secundária de uma maneira diferente. As α-hélices são desenhadas como fitas (*ribbons*), as folhas-β são desenhadas como setas e as alças e voltas como pequenos cilindros. A Figura 2.10 mostra uma representação de *Ribbons*. Esta representação usa, como pontos de controle, os átomos pertencentes ao *backbone* da proteína para desenhar estes elementos. A forma de colorir foi pela estrutura secundária;

- *New_ribbons*: esta representação é uma versão mais simplificada de *Ribbons*. Ao invés de desenhar cada elemento secondário diferente, usa os átomos do *backbone* para construir uma curva *BSpline* que passa por todos os aminoácidos, formando espirais ao longo da mesma.

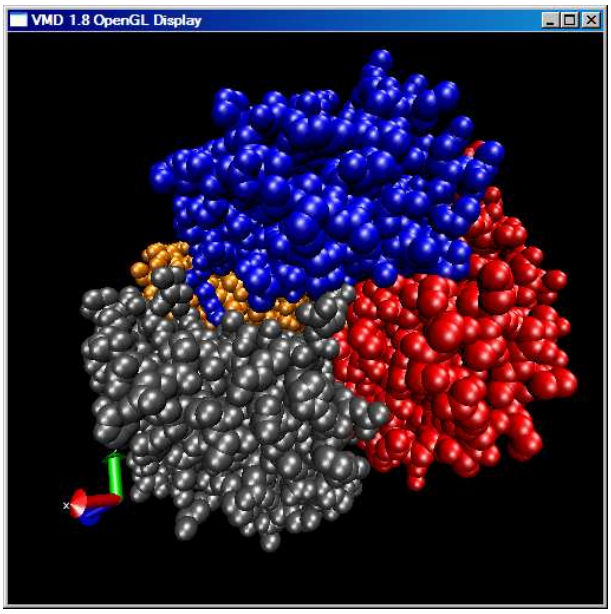

Figura 2.15: Proteína com código *PDB 1A00* desenhada com o formato *VDW* [31].

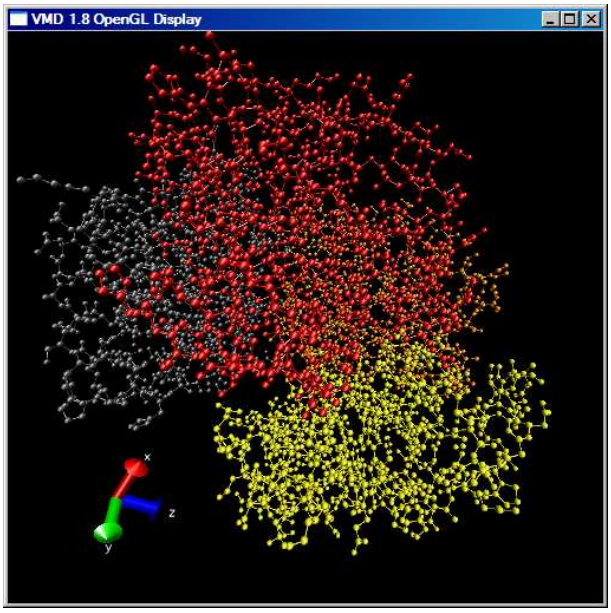

Figura 2.16: Representação da proteína com código *PDB 1A00* na forma *CPK* [31].

A Figura 2.17 mostra uma representação de *New_ribbons*, colorida pelo identificador da cadeia.

Apresentamos apenas algumas das representações de proteínas existentes. A lista é extensa e quando combinada com diferentes formas de colorir e materiais (transparências, texturas), torna a visualização das estruturas uma importante ferramenta de análise e inferência.

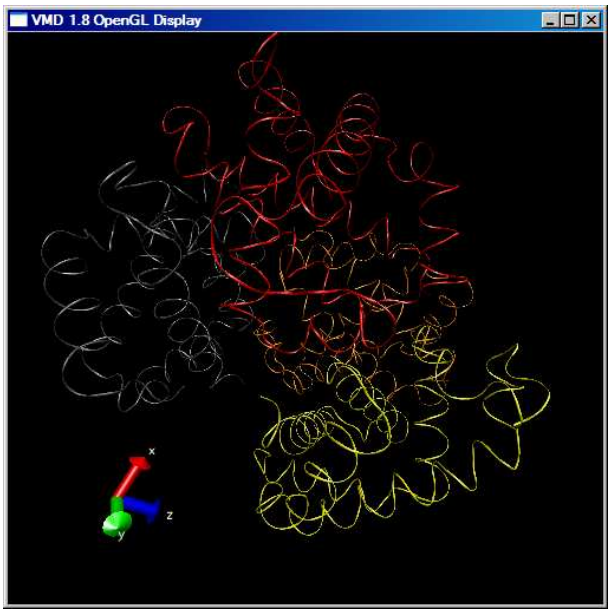

Figura 2.17: Representação da proteína com código *PDB 1A00* com a forma *New_ ribbons* [31].

2.7 Simulação de Proteínas

As simulações utilizando o método da Dinâmica Molecular, ou DM, (*Molecular Dynamics* ou MD) computam trajetórias atômicas resolvendo numericamente as equações de movimento de Newton [35]. Para o escopo deste trabalho, as simulações por DM são usadas para avaliar e quantificar a formação dos enovelamentos de proteínas e na determinação das estruturas energeticamente estáveis.

Métodos de simulação predizem as propriedades termodinâmicas de sistemas nos quais os dados experimentais não existem, são difíceis ou até mesmo impossíveis de serem conseguidos. Outro aspecto prático da realização de métodos deste tipo é sobre o monitoramento do equilíbrio de sistemas, ou seja, escolhem-se parâmetros que determinarão a parada ou a continuidade da simulação. Estas propriedades incluem a energia, a temperatura, a pressão e as propriedades estruturais [38].

Uma conformação é uma estrutura tridimensional de uma proteína. Um conjunto de conformações é chamado de trajetória, que representa a variação da posição dos seus átomos constituintes ao longo do tempo [38]. Um campo de força, no método da DM, é um conjunto de parâmetros usados para descrever as interações que ocorrem em um sistema de partículas (por exemplo, átomos).

Uma simulação gera uma trajetória como saída. A partir da variação dos parâmetros de

entrada, produz-se uma nova trajetória, para futura análise. Esta é uma característica geral de simulações: acompanhar a mudança das diferentes saídas geradas a partir da permutação dos parâmetros iniciais. Para um mesmo conjunto de átomos, este processo é repetido, formando um conjunto de dados que serve como base de comparação.

A seguir são apresentadas as ferramentas mais utilizadas para realização de simulações pelo método da DM, os parâmetros de entrada e as saídas produzidas.

2.7.1 Ferramentas de Simulação

As mais importantes ferramentas de simulação são *AMBER*, *CHARMM* e *GROMOS*:

- *AMBER* (*Assisted Model Building with Energy Refinement*): *AMBER* é uma suíte de programas para uso em modelagem molecular e simulação. Refere-se, principalmente, a dois elementos: um conjunto de campos de força para a simulação de biomoléculas e um pacote de softwares para simulação molecular, incluindo o código fonte e demonstrações [42];

- *CHARMM* (*Chemistry at Harvard Macromolecular Mechanics*): trata-se de um conjunto de ferramentas que incluem um campo de força para DM. Este pacote inclui funções de minimização energética [33];

- *GROMOS* (*GROningen MOlecular Simulation*): pacote versátil para realização de DM, simulando as equações do movimento de Newton para milhares de partículas. Apesar de ter sido originalmente concebido para biomoléculas, muitos grupos de pesquisa utilizam o GROMOS para simular sistemas não biológicos, tais como polímeros [48].

Uma simulação pelo método da DM compreende estágios que vão desde a configuração inicial de parâmetros de entrada até a produção de saídas e análise dos resultados. O objetivo final da simulação é calcular propriedades termodinâmicas das proteínas e o posicionamento dos átomos, configurando, extraindo, filtrando e interpretando as informações produzidas. Esta interpretação baseia-se no uso de ferramentas específicas de visualização e de análise (descritas no Capítulo 4).

Por fim, o *software NAMD* é um sistema para execução paralela do método da Dinâmica Molecular, desenvolvido para simulações de alto desempenho de sistemas biomoleculares de grande porte [35]. Integra-se ao *VMD - Visual Molecular Dynamics* (explicado em 4.1.3) para visualização e análise das trajetórias das simulações. O sistema é compatível com o *AMBER* e com o *CHARMM* e é distribuído gratuitamente, juntamente com o código-fonte.

2.7.2 Entradas e Saídas para o *AMBER*

Para o escopo deste trabalho, devido aos experimentos e resultados obtidos, vamos nos ater à ferramenta *AMBER*, onde estudaremos seus parâmetros de entrada e os seus dados de saída. Os parâmetros de entrada e saída, bem como todas as informações relacionadas à ferramenta estão descritas no *Manual do Usuário* do *AMBER*, versão 8.0 [2].

O *AMBER* configura parâmetros para o controle da temperatura e pressão do sistema, tempo do passo de simulação (ou *step*) e frequência com que são gerados os dados de saída para posição (arquivos no formato PDB), e saídas propriamente ditas (arquivos no formato OUT). A ferramenta cria um arquivo no formato *CRD* e um arquivo no formato *OUT* para cada tempo de simulação. Podemos destacar 3 seções importantes nestes arquivos de saída:

1. *Resource Use* (Uso dos Recursos): descreve as propriedades do sistema, como o número total de átomos e totais de ligações (*bonding totals*) com hidrogênio. Mostra também outras informações relevantes como memória usada e memória livre do sistema;

2. *Control Data for the Run* (Dados de Controle da Execução): esta seção configura parâmetros gerais do sistema tais como entrada, geração das saídas e regulagens de temperatura (para a equilibração);

3. *Results* (Resultados): mostra os resultados da execução, ou seja, exibe o comportamento da temperatura, da pressão, da energia total, cinética e potencial, para citar alguns.

Estes dados de entrada e saída são usados para análises, verificando mudanças imprevistas nos passos de simulação e estudando os dados produzidos para entender o comportamento da execução ao longo do tempo. A seguir, estudaremos as formas de análise mais empregadas para simulação utilizando o método da DM.

2.7.3 Análise de Simulações

Uma vez que a simulação executou, passa-se para a fase da análise. Uma das formas mais usuais é a geração de gráficos bidimensionais que descrevem a variação dos diferentes dados de saída ao longo do tempo de simulação. Estes são verificados à procura de ocorrências que demonstram o comportamento da conformação e as mudanças dos valores de temperatura, pressão e energia de cada passo da simulação. Além destes, outras saídas são geradas pelo *AMBER*, tais como a energia torsional e a energia eletrostática [38].

Como dito anteriormente, um aspecto da utilização de simulações tange as propriedades estruturais das conformações. A partir da Estrutura de Referência (Seção 2.5) de uma proteína calcula-se a média das distâncias entre todos os átomos da estrutura simulada e a Estrutura de Referência. Esta média das distâncias é conhecida por *RMSD - Root Mean-Square Deviation*, ou Desvio Médio Quadrático [38], como mostra a Equação 2.1.

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N_{atomos}} d_i^2}{N_{atomos}}} \tag{2.1}$$

onde N_{atomos} é o número de átomos existentes e d_i é a distância entre as coordenadas do átomo i das duas estruturas. Um gráfico do *RMSD*, em Å (ångströms), pelo tempo de simulação, em picossegundos, é exibido na Figura 2.18.

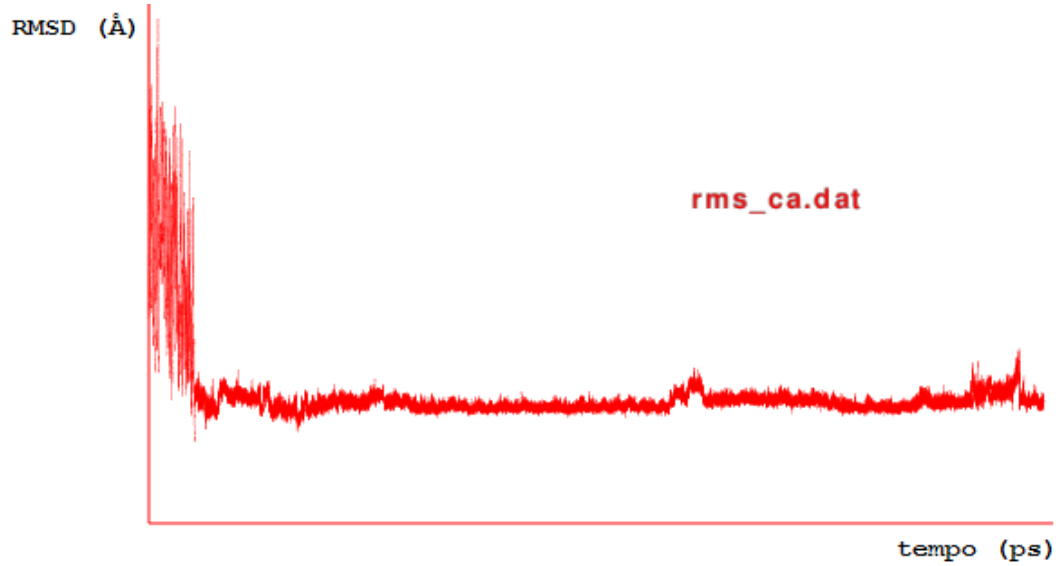

Figura 2.18: Gráfico do *RMSD* (medido em ångströms) de uma trajetória em função do tempo de simulação (medido em picossegundos) para o arquivo "rms_ca.dat" (gerado pelo *ptraj*).

A variação, no início, deve-se ao fato de que a distância em relação à estrutura de referência, além de estar variando em demasia de um passo para o outro, é alta, significando que a estrutura de estudo é bastante diferente, e, consequentemente, mais instável, se analisarmos outros atributos como energia ou temperatura.

A análise conformacional da proteína é definida como o estudo dos arranjos dos átomos no espaço tridimensional. Um componente chave deste tipo de análise é a busca conformacional, ou seja, a busca pela identificação das conformações preferidas pelas moléculas. Para esta análise são necessários métodos de minimização de energia [38].

Para análises onde é preciso estudar as conformações tridimensionais possíveis de proteínas usa-se o Gráfico de Ramachandran. Uma outra forma de análise é com o *Mapa de Contatos*. Trata-se de uma forma simples da representação estrutural de proteínas que ajuda no estudo da formação de estruturas secundárias e na escolha de funções energéticas para descoberta do estado nativo [53]. O Mapa de Contatos de uma proteína com N resíduos é uma matriz de tamanho $N \times N$ chamada S e possui os elementos definidos como:

$$S_{ij} = \begin{cases} 1, & \textit{se resíduos } i \textit{ e } j \textit{ estão em contato} \\ 0, & \textit{caso contrário} \end{cases} \tag{2.2}$$

Um Mapa de Contatos, em última análise, é um mapa de distâncias entre aminoácidos. Estas distâncias refletem a formação das estruturas secundárias na proteína. A Equação 2.2 mostra uma função para uma matriz S preenchida com o valor 1 caso exista um contato entre os aminoácidos, ou seja, existe uma distância d perto o suficiente a ponto de ser um contato. S é preenchida com 0 caso contrário.

Esta matriz binária pode ser estendida para, ao invés de guardar apenas a informação sobre ocorrência ou não de contato, guardar a distância entre os centros geométricos de cada resíduo. A Equação 2.3 define este comportamento. Seja (x_1, y_1, z_1) o centro geométrico de um aminoácido A e (x_2, y_2, z_2) o centro de um aminoácido B. A distância entre o centro de A e o centro de B é definido por:

$$D_{ij} = \sqrt{(z_2 - z_1)^2 + (y_2 - y_1)^2 + (x_2 - x_1)^2} \tag{2.3}$$

Ao criarmos uma escala de cores entre a distância mínima (que é 0, ou seja, é o próprio resíduo) e a distância máxima, temos o Mapa de Contatos definido pela Figura 2.19. As maiores distâncias (para esta proteína, este valor varia entre 22.54 e 33.90 ångströms) estão representadas por tonalidades da cor azul e, à medida que a distância diminui, é desenhada com tons de verde ou vermelho. A variação da maior distância para a menor segue o espectro de cores da luz visível, do azul ao vermelho.

Cabe ressaltar que esta matriz é simétrica, ou seja, está refletida ao longo da diagonal. Este comportamento é causado porque a distância do aminoácido A até o aminoácido B é a mesma entre o aminoácido B e o A.

Por fim, a visualização das trajetórias de simulações é um importante aspecto da análise estrutural de conformações de biomoléculas. A visualização das trajetórias gera uma animação

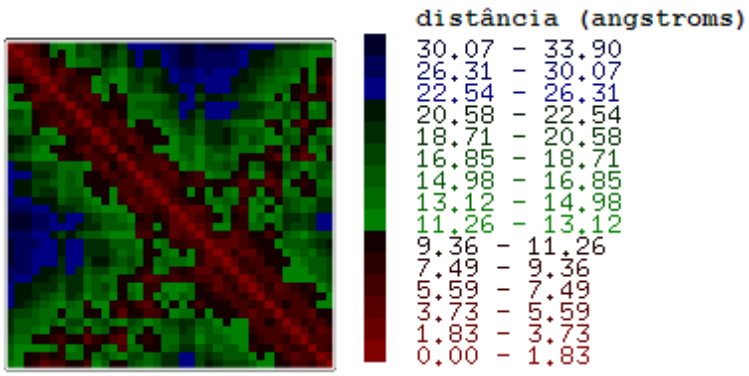

Figura 2.19: Mapa de Contatos da proteína com código *PDB 1ZDB*.

que descreve a variação estrutural da conformação e seu efeito nos atributos calculados pelo *AMBER*. A conjunção de diferentes formas de visualização com o estudo e acompanhamento dos dados de saída auxilia para a compreensão do enovelamento e na descoberta do equilíbrio dos sistemas.

Capítulo 3

Visualização e Ambientes Virtuais

Este capítulo trata sobre visualização e ambientes virtuais, listando as principais características de sistemas visuais e visualização multidimensionais de dados. Mais particularmente, estudaremos conceitos relativos a Ambientes Virtuais Ricos em Informação e Ambientes Virtuais de *Desktop* Ricos em Informação.

3.1 Visualização

A Visualização é uma área da Computação Gráfica e é considerada um suporte à tomada de decisões. Através do sistema perceptivo humano, detecta regiões de interesse, direciona a atenção e revela padrões muitas vezes escondidos. Entre os seus inúmeros benefícios está a facilidade com que os relacionamentos entre os dados são percebidos quando representados através de imagens [24, 54].

Schneiderman [49] definiu uma taxonomia a ser seguida para a construção de sistemas visuais. Estas regras são explicadas abaixo:

- Visão geral (*overview*): Mostra os dados como um todo, inclusive uma escala informando o fator de escalonamento. Neste estágio, a exibição de detalhes é restrita, possibilitando aos usuários uma visão geral;

- Filtro (*filter*): Nesta etapa, deve-se permitir que sejam retirados da cena de visualização dados sem importância. Ao remover itens, permite que o foco seja direcionado para a resolução do problema;

- Detalhes-sob-demanda (*details-on-demand*): Esta etapa permite que sejam mostrados os

detalhes para um deteminado grupo dos dados, com a finalidade de aprofundar os conhecimentos;

- Relacionar (*relate*): Ver os relacionamentos existentes entre diferentes itens. Deve permitir uma exploração de atributos similares, por exemplo, entre diferentes trajetórias de simulação;

- Histórico (*history*): Salva um histórico das ações efetuadas para suportar retorno a um estado prévio quando o usuário efetua uma alteração não desejada sobre a visualização. Esta característica faz com que os usuários identifiquem os passos necessários para gerar diferentes visualizações;

- Extração (*extract*): Permite a extração de sub-coleções e parâmetros de busca (*query parameters*). Este item determina que o sistema salve determinadas partes dos dados para análise posterior.

O *pipeline*[1] para visualização de informações foi descrito por *Card et al* [22] e é representado pela Figura 3.1. A fase inicial corresponde aos dados na sua forma mais bruta, multidimensional e heterogênea. São aplicadas tranformações nos dados para a produção de tabelas que descrevem os atributos. A seguir são construídos mapeamentos visuais, transformando os dados em estruturas visuais para a futura representação. A última transformação envolve a definição dos mecanismos navegacionais sobre as estruturas e possibilidades visuais que serão oferecidas aos usuários. Uma vez que os dados estão representados graficamente, o usuário explora este ambiente iteradamente, com o objetivo de buscar uma solução para o seu problema ou apenas navegar pelas estruturas [22, 45].

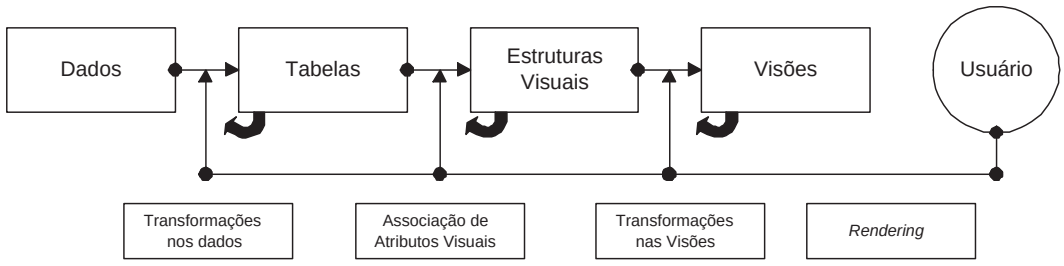

Figura 3.1: *Pipeline* para visualização de informações [45].

[1] *Pipeline* é uma metáfora para uma cadeia de estágios que transformam dados em cada passo. A entrada de cada nível corresponde à saída do anterior.

3.1.1 Visualização Multidimensional de Dados

Um tópico recorrente em Visualização versa sobre o número dos atributos para exibição, ou seja, sobre a dimensionalidade dos dados. Para muitos usuários, passar de três dimensões (3D) para quatro dimensões (4D) torna o conjunto de dados mais complexo e inacessível. Projeções acima de 4D normalmente estão além da habilidade de compreensão e são ineficazes na transmissão de informações [24]. Neste trabalho vamos estudar as visualizações multidimensionais conhecidas por *Glyphs* e *Coordenadas Paralelas*.

Glyphs são ícones que mapeiam um atributo para uma determinada forma ou símbolo que, quando visualizados em conjunto, destacam agregações e anomalias. Uma vez que um *glyph* é gerado, deve-se descobrir uma maneira de colocá-lo em uma cena de desenho. O atributo correspondente à posição do ícone é crucial para ser visualmente efetivo e comunicar padrões e relacionamentos [24, 54, 47, 52].

Este mapeamento posicional é um dos principais problemas do uso de *Glyphs*. Para lidar com estas questões existem abordagens e estratégias para lidar com esta dificuldade. O uso ou não da técnica deve ser estudado caso a caso, quando infere-se se esta é a melhor forma de representação a ser utilizada. Os métodos de desenho são variados, baseados na possibilidade de ocorrência ou não de sobreposição de itens ou na ocupação do *glyph* dentro da cena (visto que podem existir regiões vazias, dificultando a visualização) [54].

A outra forma de visualização é chamada de *Coordenadas Paralelas*. Trata-se de um método muito utilizado para representação multidimensional de informações. Resumidamente, funciona da seguinte forma: são criadas linhas paralelas verticais, correspondendo a um atributo (ou dimensão) dos dados. Estas linhas mapeiam valores unidimensionais que, ao serem conectados levando-se em consideração os múltiplos atributos, conectam pontos das diferentes dimensões, formando uma linha que mostra a variação de cada dimensão. O conjunto de todas estas linhas forma a visualização [32, 28]. A Figura 3.2 mostra uma representação de Coordenadas Paralelas com 5 dimensões.

A vantagem do uso de *Coordenadas Paralelas* é que as linhas criadas mostram a variação global da informação. A desvantagem é a demora na aprendizagem de extrações de informações pelos usuários. Esta técnica permite que sejam feitas variações, tais como redefinição das linhas paralelas (troca de ordem) para maximizar as análises [32, 28].

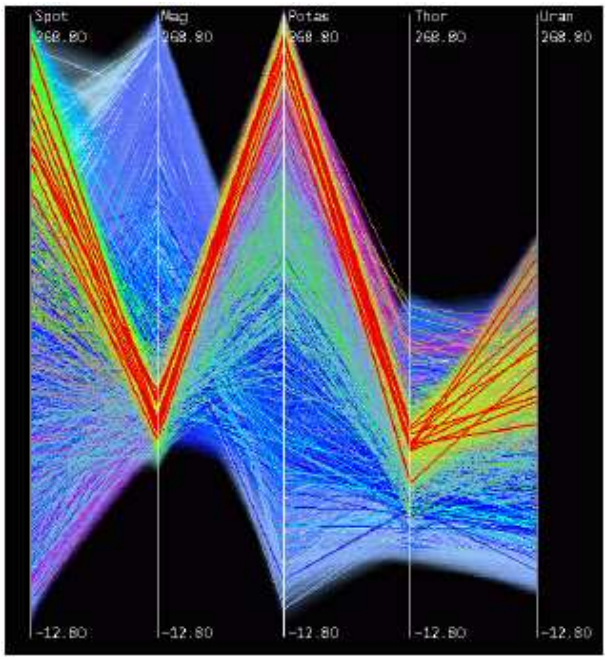

Figura 3.2: Visualização multidimensional com Coordenadas Paralelas [32].

3.2 Ambientes Virtuais

Um Ambiente Virtual (VE, ou *Virtual Environment*) é um mundo sintético, com interação em tempo real, em primeira pessoa. Uma parte importante dos Ambientes Virtuais diz respeito à interatividade com o usuário, chamada de Interação 3D (*3D Interaction*). O modo como os usuários interagem com os sistemas é crucial para a resolução de problemas e é um aspecto complicado no projeto e implementação de *softwares* por tratarem de novos conceitos, introduzindo novos problemas. O mundo real apresenta restrições e conformidades díficeis de representar com acurácia em simulações de computadores. A interação é chamada de tridimensional pois não basta aplicar os conceitos de interfaces bidimensionais para resolver os problemas, deve-se procurar estabelecer novas metáforas e formas de interação com o usuário [19].

A Interação 3D e os Ambientes Virtuais são complementares e suas áreas de aplicação são variadas, sendo usados para a criação de contextos tridimensionais realistas. Outra área de aplicação é a Visualização Científica, onde os experimentos produzem quantidades massivas de dados, e características como interpretação e *insight* são vitais para a tomada de decisão. A visualização de dados objetiva informar mais aos usuários do que puramente as tabelas contendo resultados numéricos [19].

3.3 Ambientes Virtuais Ricos em Informação

Mesclando os conceitos de análise exploratória e visualização de informações, define-se uma taxonomia para Ambientes Virtuais Ricos em Informação (*IRVE's* ou *Information-Rich Virtual Environments*) [20]. Um *IRVE* é uma combinação de um *VE* tradicional e Visualização de Informações. Sua maior contribuição é aumentar e melhorar mundos virtuais com informações abstratas [20, 19].

Visualizações tridimensionais partem de um conjunto complexo e abstrato de dados e criam representações gráficas significativas aos usuários. Estas representações são exploradas e navegadas com o intuito de descobrir mais sobre o problema que se está investigando, identificando relacionamentos e padrões nos dados. A Visualização de Informações apresenta informações abstratas usando uma forma perceptiva (normalmente uma forma visual). Já *IRVE's* **agregam** informações em um ambiente tridimensional de tempo real. Esta é a vantagem do uso de tais ambientes virtuais, pois unem percepção e informação, em um único lugar [20].

Para visualizações efetivas, pesquisadores necessitam de ferramentas integradas para a manutenção e apresentação dos dados. *IRVE's* possuem características em comum com ambientes de realidade aumentada (*AR*, ou *Augmented Reality*). Enquanto ambientes *AR's* melhoram (*augment*) o mundo físico com informações adicionais, *IRVE's* melhoram o mundo virtual com informações abstratas. Por serem puramente artificiais, *IRVE's* são mais flexíveis, pois permitem associações e ações que no mundo real não seriam possíveis [20, 46].

3.3.1 Definição formal de *IRVE's*

Por se tratar de conceitos novos em termos de ambientes virtuais, é crucial uma definição formal de *IRVE's* para estabelecer as terminologias que serão utilizadas ao longo do trabalho. *Bowman et al* [20] define uma os seguintes termos para definição de *IRVE's*:

1. *Virtual Environment*: é um mundo virtual, sintético (normalmente tridimensional) visto de uma perspectiva em primeira pessoa, controlada pelos usuários;

2. Informações abstratas: informação que normalmente não encontra-se diretamente no mundo físico. Informações sobre uma mesa (número de pés, textura) são visualmente perceptíveis, mas dados sobre a sua origem, data de fabricação são informações ditas abstratas sobre a mesa. Técnicas de visualização multidimensional são usadas para o desenho das informações abstratas (algumas técnicas estão descritas na Seção 3.1.1);

3. Realismo em *VE's*: um *VE* é dito **realístico** se seus componentes perceptíveis representam componentes igualmente perceptíveis no mundo físico. Caso contrário, o *VE* não é realístico e sim **abstrato**. Um bom exemplo deste conceito é um monumento (existente em um determinado lugar), ou uma molécula (que existe em uma escala reduzida) ou uma casa. Todos estes componentes podem ser considerados realísticos, mas cubos em uma cena tridimensional representando três diferentes atributos de um sistema são considerados componentes abstratos;

4. *IRVE's*: um *IRVE's* é um *VE* realístico quando é melhorado com a adição de informações abstratas **relacionadas**;

5. Fidelidade da percepção do mapeamento da informação: esta questão abrange a preocupação de quão fiel um *IRVE* representa as informações do mundo físico que são percebidas no mundo virtual. Em alguns casos, informações são **alteradas** para mostrar novas informações abstratas em objetos, enquanto que em outros casos informações são **adicionadas** ao ambiente sem alteração em termos de percepção do ambiente original;

6. Visualização de Informações 'Pura': uma visualização com *Coordenadas Paralelas* de dados de pacientes, por exemplo, **não** é um *IRVE* pois todas as informações do ambiente são abstratas, mapeadas para a forma perceptual. Um *IRVE* é caracterizado por **adicionar** informações relevantes.

As definições formais descritas fundamentam os conhecimentos sobre *IRVE's* para o projeto, definição arquitetural, concepção e implementação de sistemas de visualização e análise. Dando seguimento a outras questões pertinentes à *IRVE's*, na próxima seção vamos discorrer sobre como disponibilizar informações em cenas tridimensionais.

3.3.2 Especificações para o Tratamento das Informações

A questão de onde colocar os elementos na cena tridimensional é relevante no estudo de *IRVE's*. A área dos Ambientes Virtuais estuda a representação que é percebida pelos usuários, enquanto a área da Visualização de Informações lida com os dados abstratos. Novas técnicas são necessárias para unir estes conceitos.

Uma das formas de disponibilização de informações é através do uso de painéis informativos. São superfícies bidimensionais normalmente transparentes onde são desenhados os atributos dos objetos ou da cena [44].

As principais preocupações no projeto de *IRVE's* são sobre a localização do desenho, a associação das informações abstratas, visibilidade, legibilidade, oclusão e o nível de agregação [20, 44]. Estes conceitos serão melhor estudados a seguir:

- Localização do Desenho (*Display Location*): esta especificação é baseada em conceitos definidos em realidade aumentada. A área de desenho é dividida em **fixa ao mundo** (*world-fixed*), **fixa à cena de desenho** (*display-fixed*), **fixa ao objeto** (*object-fixed*) e **fixa ao usuário** (*user-fixed*). Quando a informação é atrelada a uma localização tridimensional específica no ambiente virtual, esta é dita *fixa ao mundo*. As informações que permanecem na mesma posição da cena de desenho (tela) são conhecidas como *fixas à cena de desenho*. São ditas *fixa ao objeto* quando as informações são associadas aos objetos do sistema, e movimentam-se de acordo com a posição do objeto. Por fim, as *fixas ao usuário* acompanham os movimentos e a navegação do usuário;

- Associação (*Association*): a associação é relativa aos relacionamentos existentes entre informações abstratas e perceptíveis. A representação destas relações podem ser *espaciais* ou *visuais* e *implícitas* ou *explícitas*. As espaciais referem-se à posição dos objetos. Quando não existem relações aparentes, mas uma determinada seleção de um objeto implica na alteração visual (um destaque como, por exemplo, alteração de cor) de outro objeto, em outro lugar, é dita visualmente implícita. Mas se um objeto possui um gráfico associado e este é conectado através de uma linha na cena de desenho, esta associação é dita explícita;

- Nível de Agregação (*Level of Agregation*): Trata do quanto de informações abstratas podem ser visualizadas ou agregadas e dispostas em representações mais complexas (talvez usando uma composição de diferentes representações). Caso seja escolhida a forma separada, facilita o estudo de detalhes dos objetos (a localização da informação mais indicada para este caso seria a *fixa ao objeto*). Na forma agregada, aumenta a flexibilização em escolher diferentes tipos de visualização de informações, e a forma de localização seria *fixa à cena de desenho* ou *fixa ao usuário*;

- Visibilidade: Os painéis informativos devem ser visíveis aos usuários. A primeira consideração a ser observada é o tamanho da anotação (ou descrição). A uma determinada distância, o painel deve ser suficientemente grande para ser percebido, mas não tão grande que domine o campo de visão, sendo percebido como um objeto na cena e não como um painel que mostra alguma informação sobre algum objeto;

- Legibilidade: Quando a informação apresentada não está legível, a fonte utilizada não é apropriada, seu tamanho é muito pequeno ou a cor não é bem escolhida. Os usuários devem poder ler a informação que é apresentada na cena, tornando a escolha dos atributos da fonte uma questão chave para representação de textos e gráficos;

- Oclusão: a oclusão é o problema que acontece quando muitos painéis são criados e mostrados aos usuários, tornando a cena confusa e consumindo por completo o espaço visual. Estes problemas são resolvidos com gerenciadores de painéis, que detectam o número de painéis abertos ou a área livre possível de desenho. Esses gerenciadores definem regras para configurar o que é mostrado para o usuário de forma que não sobrescreva informações (impossibilitando a visualização).

Quando as informações abstratas são referentes ao mundo ou ao objeto, duas formas podem ser usadas: associar a informação adicionando uma representação sensorial (um objeto visível, por exemplo). Os objetos são vistos através de *Glyphs*, gráficos, texto ou outra representação. Uma diferente forma é associar a informação através da mudança dos objetos no ambiente virtual, por exemplo, cores mais vívidas representando o custo elevado para a construção de uma parede em uma casa [20].

Além da disponibilização, questões relativas especificamente ao leiaute dos textos também são necessárias. Uma classificação destas técnicas é estudada na próxima seção.

3.3.3 Classificação de Técnicas de Leiaute de Textos

Os textos inseridos na cena são classificados quantitativamente (pela quantidade de informação agregada), qualitativamente (baseado pela localização espacial) e temporalmente (modificam o desenho em função de uma escala temporal). *Bowman et al* [20] define uma taxonomia para refletir categorias de técnicas de leiaute de textos. Trata-se de uma base teórica para implementações de sistemas deste porte. A Figura 3.3 mostra a classificação em um alto nível de abstração.

Segundo esta classificação, simples rótulos (*labels*) são considerados baixos em termos de conteúdo textual. Da mesma forma, descrições de elementos na cena são vistos como tendo qualidade média e descrições com maior número de detalhes como detentores de qualidade alta. Os atributos visuais do texto são os aspectos que afetam diretamente a legibilidade do texto e a percepção global no ambiente virtual. Os atributos incluem o tamanho da fonte, sua cor, tipo e

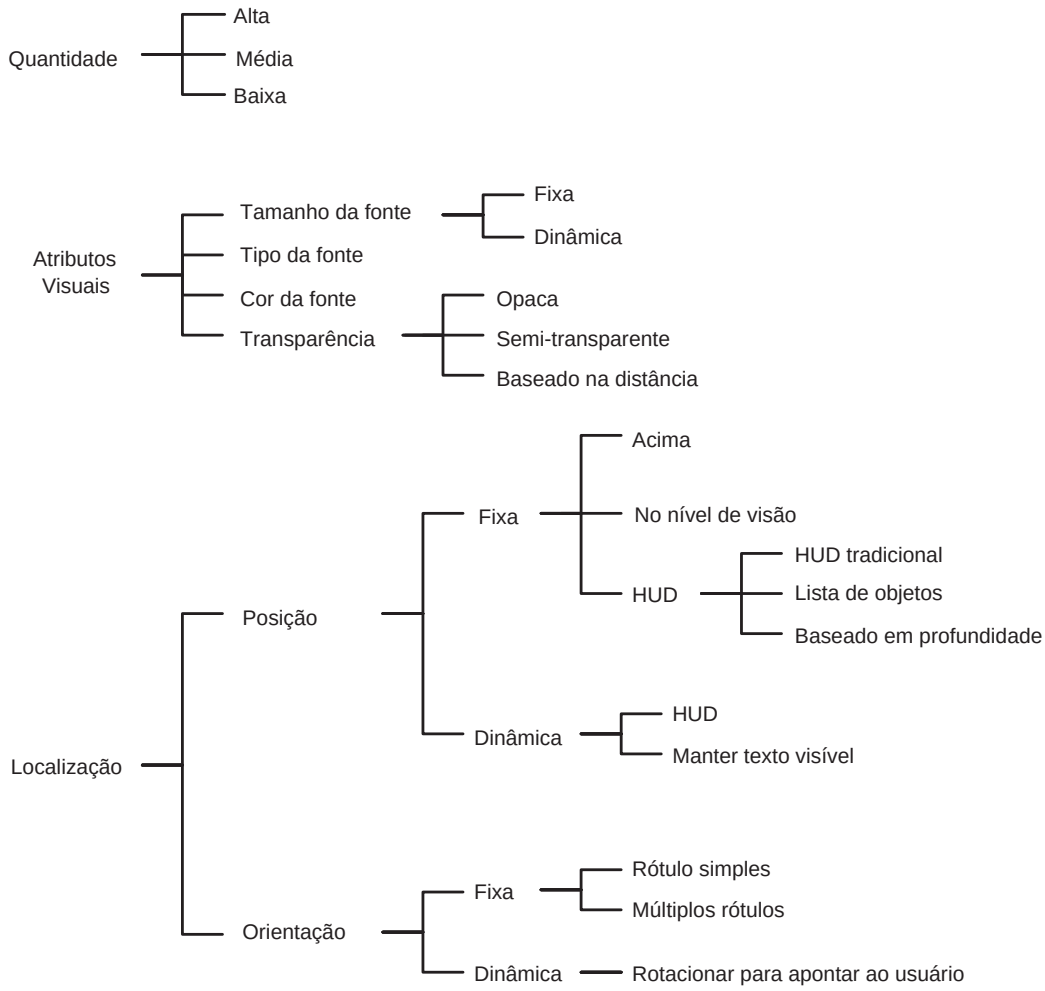

Figura 3.3: Taxonomia de técnicas de leiaute de textos para *IRVE's* [20].

transparência do texto.

O tamanho do texto pode ser fixo ou variável, com a finalidade de refletir alguma propriedade ou dimensão do sistema. Esta variação dinâmica pode ser baseada na distância do usuário ao objeto ou em outro aspecto do ambiente. A transparência refere-se ao painel no qual o texto é disposto (e não ao texto propriamente dito). Esta região (uma caixa, um retângulo ou outra forma) pode ser opaca ou semi-transparente, sendo que pode ocorrer uma variação da transparência também de acordo com a distância do usuário.

A posição do texto pode ser fixa ou dinâmica em relação ao espaço, de acordo com a visão do usuário. Quando o texto é fixo, a forma usual é posicioná-lo acima do objeto, fixo na mesma linha de visão do usuário ou fixo ao HUD (*Heads-up Display*).

A orientação do texto também é disposta em duas categorias: fixa e dinâmica. Na fixa, usa-

se rótulos singulares (como uma *billboard*, ou seja um painel onde as informações são dispostas) e o uso de múltiplos rótulos (por exemplo, um cubo no qual todas as faces possuem o mesmo conteúdo textual). Na dinâmica, o texto é rotacionado para sempre apontar para o usuário. *Bowman et al* [20] não define classificações para textos temporais, mas existem possibilidades do uso de animações e placas ou marquises informativas [20, 44].

3.3.4 Objetos Semânticos

Bederson et al [15] propõe que os projetistas de interface usem o conhecimento do usuário sobre o mundo real, por exemplo, objetos aparecem e se comportam diferentemente, dependendo da escala da visão e o contexto. Estes aspectos culminam com a proposição de uma "física de interface", chamada *Aproximação Semântica* (em inglês, *Semantic Zooming*), onde tanto o conteúdo da representação e sua manipulação são disponibilizadas diretamente e naturalmente aos usuários [20].

Essa definição ajuda na compreensão dos *Objetos Semânticos*, os quais são representados diferentemente dependendo da distância à qual o objeto é percebido pelo usuário, considerando um ambiente virtual onde existem diversas informações abstratas e heterogêneas associadas com alguns objetos da cena tridimensional. De uma certa distância, apenas um título genérico seria mostrado ou o nome de cada objeto. À medida que o usuário navegasse para perto dos objetos, seriam mostradas outras informações sobre os objetos tais como descrições, outros rótulos e detalhes (a aproximação determinaria que agora estes elementos podem ser vistos) [20].

3.4 Ambientes Virtuais de *Desktop* Ricos em Informação

Além do formalismo definido pelos *IRVE's*, foram definidos os chamados Ambientes Virtuais de *Desktop* Ricos em Informação (*Desktop Information-Rich Virtual Environments*) que são VE's que rodam na tela do computador e que não precisam de nenhum equipamento específico para serem operados. Segundo *Polys* [44], estes VE's são mais flexíveis pois utilizam conceitos tais como múltiplas janelas externas. Desta forma, correspondem à exibição de informações paralelas relacionadas ao mesmo tempo que são exibidos na cena tridimensional.

Os desafios específicos no projeto de *Desktop IRVE's* relacionam-se ao limite restrito de espaço visual disponível (a tela de um computador, por exemplo). Em contextos de *desktop*, onde múltiplas janelas são viáveis para a apresentação de informações complementares, o mais impor-

tante é estabelecer uma correspondência perceptual entre os objetos no ambiente tridimensional e os itens nas outras áreas. Esta correspondência pode ser atingida com o uso de cores, onde os elementos com a mesma cor compartilham de uma ligação própria, significando que estão relacionados de alguma maneira [44].

Os *Desktop IRVE's* antecipam problemas relativos à visualização e interação com as informações paralelas dispostas para os usuários. As associações entre o objeto 3D e as janelas abertas (normalmente contendo conteúdos bidimensionais, na forma de gráficos ou textos complementares) devem trocar informações e atualizarem-se de acordo com a necessidade. Se um usuário seleciona uma informação específica num contexto bidimensional, o objeto correspondente na cena tridimensional deve ser igualmente selecionado e vice-versa [44].

Capítulo 4

Trabalhos Relacionados

O objetivo deste capítulo é comparar as ferramentas atuais de visualização e análise de trajetórias de simulações. As ferramentas de visualização escolhidas foram o *PyMOL Molecular Graphics System* [25], o *DeepView - Swiss PDB Viewer* [29] e o *VMD - Visual Molecular Dynamics* [31]. As ferramentas de análise estudadas foram o *ptraj/rdparm* (do pacote de simulação *AMBER* explicado na Seção 2.7.1) onde descreveremos outros *softwares* que complementam as informações disponibilizadas aos usuários.

4.1 Sistemas de Visualização de Proteínas

Atualmente, a lista de sistemas de visualização de proteínas é extensa e cada ferramenta ajuda na resolução de uma parte da análise. A seguir, uma descrição de alguns dos sistemas mais utilizados por profissionais da bioinformática para visualizar proteínas.

4.1.1 *PyMOL Molecular Graphics System*

O *PyMOL Molecular Graphics System*, versão 0.98 (Maio de 2005), ou simplesmente *PyMOL*, é disponibilizado pela empresa *DeLano Scientific*. Esta versão é distribuída para sistemas operacionais *Linux*, *MacOS* e *Windows*. A linguagem de programação utilizada foi *Python*. Trata-se de um sistema gráfico de apresentação de moléculas, com um interpretador *Python* embutido, desenvolvido para visualizações em tempo real, geração rápida de imagens e animações moleculares de alta qualidade [25, 11].

O *software* foi lançado para se tornar uma alternativa viável à sistemas comerciais de alto custo e recebe como entrada arquivos no formato *PDB*, *Macromodel*, *SDF* e *MOL* [11]. Suas

principais funcionalidades incluem: visualização de estruturas tridimensionais otimizadas; oito representações moleculares diferentes; cálculo de distâncias e anotações; exibição de superfícies transparentes; linha de comando interativa (*Command Line Interface*); superposição de moléculas; animação de estruturas; desenvolvimento modular e orientado a objetos.

A maior desvantagem do *PyMOL* é não abrir trajetórias de simulação. A Figura 4.1 mostra a interface gráfica do sistema. A proteína com código *PDB 1A00* foi representada com *Ribbons* e colorida de acordo com a estrutura secundária (na cor vermelha para as *alpha*-hélices e na cor verde para as alças e voltas.

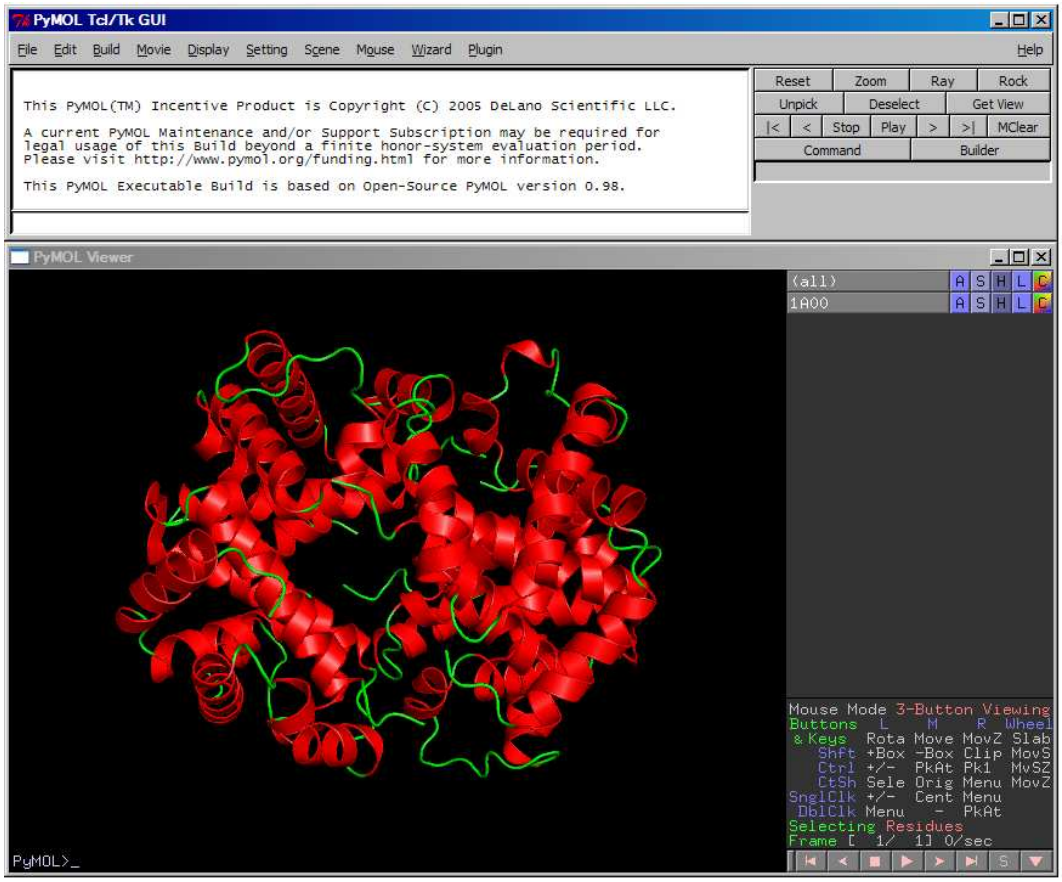

Figura 4.1: Interface gráfica do sistema *PyMOL* da proteína com código *PDB 1A00*.

4.1.2 *DeepView - Swiss PDB Viewer*

O *software DeepView - Swiss PDB Viewer* [29], ou apenas *DeepView*, versão 3.7, é disponibilizado pelo *Swiss Institute of Bioinformatics* (*SIB*) em colaboração com o departamento de Pesquisa e Desenvolvimento da *GlaxoSmithKline* e o *Structural Bioinformatics Group* situada

em Basel, Suíça.

O *DeepView* é portável para as plataformas *MacOS*, *Windows*, *SGI* e *Linux*. O código-fonte é proprietário e não está disponível para alterações ou estudos. O *DeepView* é uma aplicação com uma interface gráfica amigável ao usuário, usada para a visualização de múltiplas proteínas de forma simultânea (na mesma cena de desenho). Este *software* foi construído com a biblioteca gráfica *OpenGL* e abre arquivos do formato *PDB*. Além das representações usuais, calcula distâncias e ângulos.

A Figura 4.2 demonstra a interface gráfica do *DeepView*. A proteína com código *PDB 1A00* foi desenhada com a representação *Ribbons*, e a forma de colorir foi pelo identificador da cadeia. À direita, o sistema utiliza uma janela auxiliar para selecionar aminoácidos, escolher representações e formas de colorir, além de aplicar filtros em elementos específicos (mostrar/esconder aminoácidos, grupos de aminoácidos e cadeias).

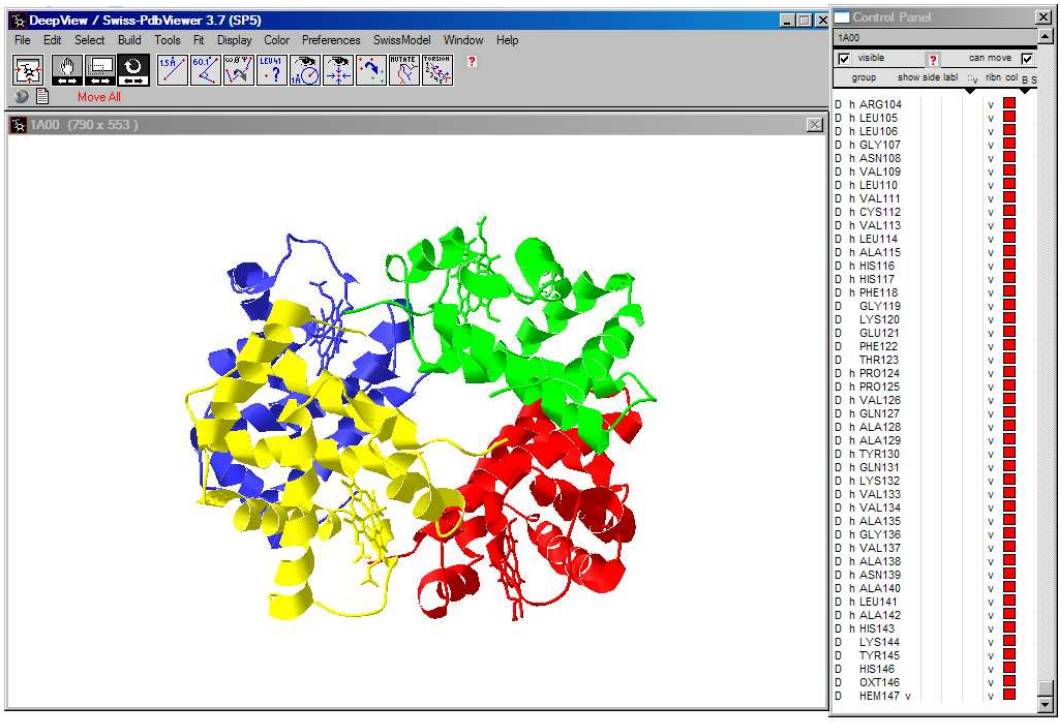

Figura 4.2: Proteína com código *PDB 1A00* representada no sistema *DeepView*.

4.1.3 *VMD - Visual Molecular Dynamics*

O *Visual Molecular Dynamics*, ou apenas *VMD* [31], versão 1.8.2 é financiado pelo *National Institutes of Health* dos E.U.A. e localizado no *Beckman Institute* da Universidade de Illinois em Urbana-Champaign. As plataformas que suportam o *VMD* são a *AIX*, *HPUX*, *IRIX*, *Linux*, *Macintosh*, *Solaris* e *PC*. Sua implementação foi realizada com a linguagem C/C++ e Tcl/Tk para a interface gráfica.

O *VMD* é uma ferramenta desenvolvida para a visualização e análise de sistemas biológicos, tais como proteínas, ácidos nucléicos e lipídios. Este sistema foi construído com a biblioteca gráfica *OpenGL*. A instalação de *plugins* aumentam a lista de formatos aceitos, além de estender as suas funcionalidades. O sistema é comumente adotado para visualização, estudos para refinamento de estrutura, análise de trajetórias e estudos interativos de dinâmica molecular.

Trata-se de um *software* fácil de usar, com código-fonte aberto. O sistema é usado para animar e analisar trajetórias de simulações, atuando como um cliente (*front-end*) visual de uma outra ferramenta de simulação executando em uma estação remota (por exemplo, a ferramenta *AMBER*, descrita na Seção 2.7.1); abre dados moleculares de grande porte; integra-se ao *software STRIDE* (Seção 2.3).

Suas funcionalidades principais são: animação de estruturas moleculares; criação de diversas representações para uma estrutura (inclusive múltiplas representações para diferentes partes da proteína). Para um uso mais avançado do sistema, exige o conhecimento de linguagens de *script* (no caso a linguagem *Tcl/Tk*). Os *scripts* também são usados para personalizar representações e tratar dados de simulações.

A Figura 4.3 mostra a interface gráfica do *VMD* com a proteína com código *PDB 1A00*. O filtro utilizado foi por *backbone* (ou seja, o sistema retirou da representação as cadeias laterais dos aminoácidos) e está colorindo a proteína de acordo com o nome do aminoácido (conforme demonstrado pela Figura 4.3a). Na Figura 4.3b, temos a janela de representação, mostrando as opções definidas para visualização da proteína. A Figura 4.3c indica a janela principal do sistema, listando o nome da proteína que foi aberta e o seu número de átomos.

4.2 Sistemas de Análise de Trajetórias de Simulações

Além dos sistemas de visualização, apresentaremos *softwares* específicos para complementação das análises de trajetórias de simulações.

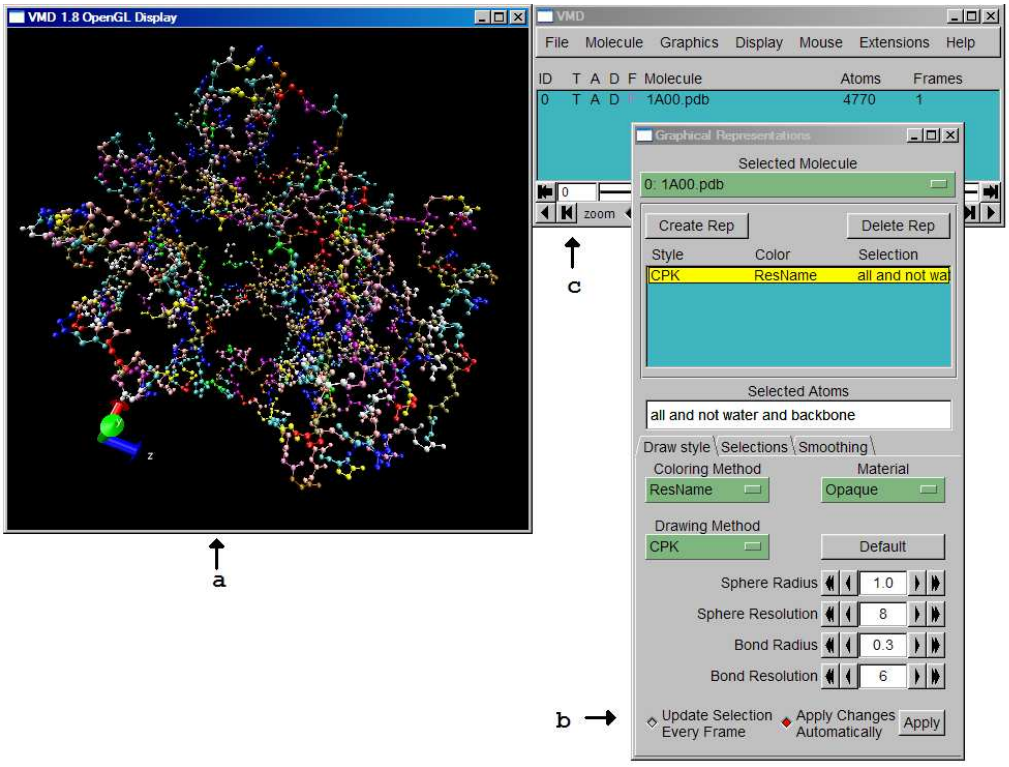

Figura 4.3: Interface gráfica do *VMD* da proteína com o código *PDB 1A00*. Em a) Janela de visualização. b) Janela auxiliar de representações gráficas e c) Janela principal do sistema.

4.2.1 *ptraj/rdparm*

São programas escritos na linguagem de programação *C* que desempenham funções de leitura (*rdparm*) e processamento (*ptraj*) de arquivos gerados pelo *AMBER*, pelo *CHARMM* ou simplesmente arquivos no formato *PDB*. Suas funcionalidades principais são: converter trajetórias para outros formatos de arquivo, cálcular ângulos, ligações atômicas(*bonds*), valores de *RMSD* e flutuações posicionais de átomos [2].

O objetivo principal do *ptraj* é processar trajetórias. Para cada conjunto de coordenadas de uma trajetória, uma sequência de eventos e ações é realizada (segundo uma ordenação de eventos definidas pelo usuário), gerando um conjunto de dados de saída. Os cinco passos específicos para o *ptraj* são [2]:

1. leitura do arquivo de parâmetros ou topologia: nesta etapa é feita a configuração inicial do sistema. A informação contida nesses arquivos trata sobre o número de átomos, aminoácidos, nomes de átomos e nomes de aminoácidos;

2. configuração da lista de arquivos de coordenadas: especifica o nome do arquivo de coordenadas e, opcionalmente, um valor inicial, um valor final e um *offset* de leitura, facilitando as tarefas caso a execução da simulação parou abruptamente;

3. especificação do arquivo de saída (tarefa opcional): configura a forma de geração dos arquivos de saída;

4. especificação de uma série de ações a serem seguidas, para cada conjunto de coordenadas: esta etapa é usada para configurar opções de análise e manipulação de coordenadas, onde as tarefas são executadas sequencialmente;

5. geração/análise de resultados (tarefa opcional): nesta etapa são produzidas as informações de saída.

O *ptraj/rdparm* é um *software* de linha de comando, recebendo arquivos no formato texto como entrada e gerando outros arquivos no formato texto como saída. O *software* é distribuído juntamente com o *AMBER* e é uma ferramenta útil para análise de trajetórias.

4.2.2 Outros *Softwares* e Bibliotecas

Para que os usuários façam outras análises sobre os dados, existem sistemas auxiliares que complementam as informações, gerando gráficos e calculando estatísticas.

Para a leitura de diferentes formatos de arquivos, uma alternativa de uso é a biblioteca *OpenBabel*. Trata-se de um conjunto de funções para abrir e converter muitos dos formatos utilizados em bioinformática, física e química. É uma biblioteca com código-fonte aberto e sem custos de licença.

Para análises estruturais de proteínas, um importante *software* é o *PROCHECK* [37]. São programas computacionais que atestam a qualidade estrutural de proteínas, gerando o *Gráfico de Ramachandran*, calculando análises estruturais do *backbone* e propriedades dos resíduos.

O *software Origin* é uma alternativa para o desenho de gráficos a partir das saídas do *ptraj/rdparm*. É um sistema proprietário que oferece 60 formas de desenho de gráficos bidimensionais, tridimensionais e representação de contornos. O *software* agrega ferramentas de análise, como análises estatísticas, processamento de sinais e geração de curvas [8].

4.3 *Virtual Data Visualizer*

O *Virtual Data Visualizer*, ou *VDV*, é um conjunto de ferramentas para visualização exploratória de dados. Sua principal característica é não se focar em um conjunto específico e dados, mas sim servir a um conjunto mais genérico e amplo [52]. O *VDV* é uma aplicação interessante sobre a integração de visualização de dados e análise.

Uma de suas funcionalidades é explorar dados multidimensionais, onde os usuários são colocados em um ambiente onde primeiramente navegam pelos dados, sem nenhum objetivo específico. O nome desta modalidade é Visualização Exploratória [24, 52]. O *VDV* baseia-se no uso de *Glyphs* para representação multidimensional (o uso destes elementos visuais está descrito na Seção 3.1.1).

4.4 *PathSim Visualizer*

O *PathSim Visualizer* é um IRVE (estes ambientes estão descritos no Capítulo 3) específico para a análise e visualização de resultados de simulações de agentes patogênicos, onde os usuários interagem e complementam os dados. O objetivo é agregar à visualização anotações e informações abstratas, intensificando a experiência de interpretação que os usuários realizam sobre os dados [46].

O sistema foi desenvolvido na linguagem *VRML* (*Virtual Reality Modelling Language*) e *XML* (*Extensible Markup Language*) para guardar e recuperar as informações. O projeto do sistema usa três diferentes tipos de informações: informações espaciais e multi-escalares, informações abstratas e informações temporais. A primeira diz respeito às diversas escalas de informações provenientes de dados anatômicos. O sistema visualiza macro e micro escalas, detectando a proximidade do usuário em relação a um objeto de estudo.

As informações abstratas tratam de informações relevantes, mapeadas dentro do sistema, tais como índices para populações de vírus, anotações, *hiperlinks* e referências sobre a estrutura estudada. Por fim, informações temporais lidam com o aspecto dinâmico e temporal das informações abstratas e espaciais inseridas na cena de desenho.

4.5 Discussão dos Trabalhos Relacionados

O conjunto dos sistemas especializados criam um ambiente para a resolução de problemas. A união de todas as informações produzidas pelos diferentes sistemas formam uma análise completa. Estas informações são úteis para a tomada de decisões quanto à qualidade da trajetória da simulação.

O maior problema que encontramos nas ferramentas estudadas acima são relativos às linguagens de *script*. Apesar de serem um forma poderosa de dinamicamente alterar características das visualizações e realizar análises, são maneiras complicadas de interação com o usuário, que, no caso, são profissionais da bioinformática (e, muitas vezes, usuários destreinados e que não sabem programar).

O *Deep View* não tem o código-fonte aberto e não pode ser usado para análises de trajetórias, apesar de ser eficiente e útil. Já o *PyMOL*, tem uma interface gráfica de difícil interação e também não abre arquivos de trajetórias. Dentre os sistemas de visualização, o melhor sistema avaliado foi o *VMD* pois tem o código-fonte aberto (e livre de custos de licenças), abre trajetórias de simulações de formatos diferentes e é simples de usar.

A principal dificuldade das ferramentas é resolver problemas exclusivos de visualização ou análise. Esta separação exige que os usuários saibam usar e compreender diferentes ferramentas, normalmente situadas em diferentes plataformas. É desta problemática que surge a necessidade da criação de um ambiente integrado e informativo, que trata da convergência entre sistemas puramente visuais e puramente de análise.

O *PathSim Visualizer* é um projeto de ambiente informativo interessante e que define algumas questões de projeto que são relevantes. Entretanto, trata-se de um sistema escrito na linguagem *VRML* e *XML* e é específico para a resolução de simulações patogênicas.

Já o *VDV* usa *Glyphs* para visualização multidimensional. A idéia é válida, entretanto, seria melhor utilizado caso oferecesse outras opções de visualização multidimensionais aos usuários. O problema do *VDV* é que ele serve para múltiplas fontes de dados (dados atmosféricos, de engenharia e física), ou seja, é genérico. Em muitos casos, *Glyphs* não resolvem os problemas de cada caso específico.

A característica principal desses ambientes é proporcionar aos usuários uma melhor experiência de interpretação dos dados e de resolução dos problemas, agregadas em uma mesma ferramenta. O Capítulo 5 tratará sobre este aspecto, onde definiremos um sistema tanto visual, quanto de análise, utilizando os conceitos de *IRVE's* definidos no Capítulo 3.

Capítulo 5

O Ambiente *SimVIZ*

Após o estudo de *IRVE's* descritos na Seção 3.3 e com base nos trabalhos relacionados, podemos agora formalizar o Ambiente *SimVIZ* apresentando suas características, módulos, diagrama de classes e principais funcionalidades.

A Seção 5.1 descreverá o Ambiente *SimVIZ* e apresentará um fluxograma básico do processo de abertura de uma trajetória de simulação, para, na Seção 5.2 explicar os módulos principais do sistema. A Seção 5.3 descreverá como os elementos gráficos colaboram para a criação do ambiente e como as visualizações e representações podem ser combinadas ou removidas da cena de desenho. Na Seção 5.4 será descrita a arquitetura do *SimVIZ* e na Seção 5.5 serão discutidos os *feedbacks* com os usuários. Por fim, a Seção 5.6 trata sobre as considerações finais sobre o ambiente.

5.1 Descrição do Ambiente *SimVIZ*

O Ambiente *SimVIZ* baseou-se em conceitos de Ambientes Virtuais Ricos em Informação (*IRVE's*), mais especificamente nos Ambientes Virtuais de *Desktop* Ricos em Informação, pois não estamos trabalhando em um contexto de realidade virtual imersiva. Estes contextos possuem características diferentes dos ambientes de aplicações de *desktop*. O principal objetivo do Ambiente *SimVIZ* é integrar a visualização das trajetórias com a análise, através da disponibilização de diferentes formas de representações gráficas, como o Gráfico de Ramachandran, Mapa de Contato, gráficos de saídas de simulações e gráficos de *RMSD* [21, 53, 38].

Além destas formas usuais de análise e visualização, o ambiente permite que sejam associadas outras informações relevantes na cena de desenho com a finalidade de ampliar os conhecimentos

prévios sobre a dinâmica da trajetória e verificar o comportamento da simulação ao longo do tempo. Para atingir estes objetivos, esse ambiente abre arquivos de coordenadas atômicas e saídas de simulações (geradas previamente pelo *AMBER*), constrói a topologia da proteína (usando informações dos aminoácidos), exibe proteínas e detalhes específicos (elementos químicos, estruturas secundárias), oferece a visualização de informações associadas a estas simulações (também para os passos de simulação e aminoácidos) e permite interações tais como rotações, translações e escalas de objetos gráficos.

A Figura 5.1 mostra o fluxograma do *SimVIZ*, desde a abertura dos arquivos de coordenadas até a visualização na cena de desenho. O fluxograma explica o processo para abrir uma trajetória de simulação. No ínicio, o usuário escolhe abrir um arquivo de coordenadas de proteínas, ou seja, um arquivo no formato *PDB*. Depois de escolher o arquivo, o sistema calcula as distâncias entre os aminoácidos e executa o *software STRIDE* que determina as estruturas secundárias. A próxima tarefa é adicionar a proteína na lista de proteínas que o sistema guarda e detectar se o usuário está abrindo uma trajetória de simulação ou apenas uma conformação ou estrutura (isso depende de onde o usuário escolheu abrir a proteína na interface). Se for uma simulação, o *SimVIZ* abre o arquivo das saídas do *AMBER* e verifica se ainda existem outras conformações ou instantâneos da trajetória para serem abertas. Se existirem, repete este processo para o próximo tempo de simulação até todos estes serem processados.

Se não existirem novos tempos de simulação para serem abertos ou não se tratar de uma simulação, o ambiente abre as *Informações da Sessão* (informações provenientes do enriquecimento que foram associadas pelo usuário anteriormente), realiza as contagens e cálculos de médias, desvios e valores máximos e mínimos para as saídas de simulação, bem como monta a topologia dos aminoácidos da proteína. Após o término desta tarefa, o sistema constrói uma visualização padrão inicial para a conformação do primeiro passo de simulação e a desenha na cena tridimensional de desenho, onde fica à espera de comandos do usuário para alteração de representações, escalas e outras funcionalidades, descritas ao longo deste capítulo.

Cabe ressaltar que este fluxograma trata da abertura de uma trajetória e, consequentemente, uma janela de visualização. Caso o usuário queira abrir outras trajetórias, basta repetir esse processo.

Começando por um nível mais abstrato em termos de descrição do ambiente vamos nos concentrar na explicação dos dados que estamos usando para a produção dos resultados.

Trabalharemos com uma trajetória de simulação com as seguintes características:

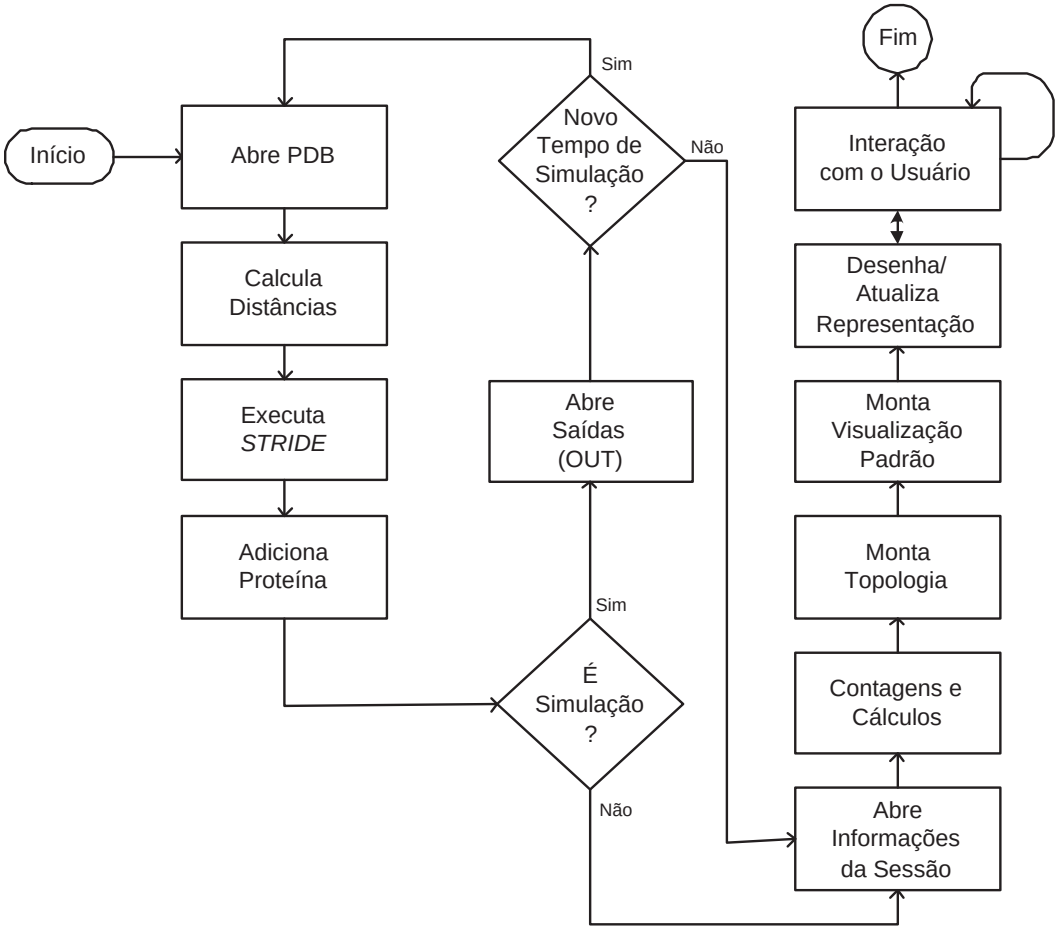

Figura 5.1: Fluxograma inicial de execução do Ambiente *SimVIZ*.

- proteína de referência: código *PDB 1ZDB*;
- trajetória de simulação por MD para predição de estrutura com duração de 100 nanossegundos;
- 100 arquivos no formato *PDB*, um para cada nanossegundo do tempo de simulação, contendo as posições atômicas dos átomos da proteína;
- 100 arquivos no formato *OUT*, um para cada nanossegundo do tempo de simulação, contendo as saídas da simulação;
- 4 arquivos de *RMSD* gerados pelo *ptraj* (Seção 4.2.1): *rms_ca.dat* (*RMSD* usando os Carbonos-α como referência), *rms_ca_2-32.dat* (dados dos aminoácidos 2 ao 32), *rms_helI.dat* (dados para a hélice 1), *rms_helII.dat* (dados para a hélice 2). Estas hélices estão mostradas na Figura 5.2.

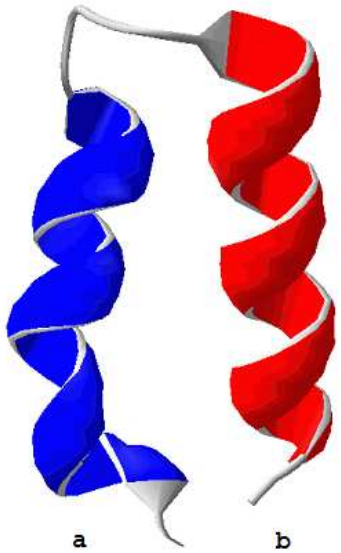

Figura 5.2: Proteína da Estrutura de Referência com código *PDB* 1ZDB. Em (a) hélice 1 e (b) hélice 2.

A seguir descrevemos os módulos do ambiente e como resolvemos os problemas de leitura dos dados, montagem de topologias, enriquecimento de informações e produção de saídas gráficas bidimensionais e multidimensionais.

5.2 Módulos do Ambiente *SimVIZ*

O *SimVIZ* é executado através de etapas distintas desde a abertura de arquivos e mapeamentos até a visualização das múltiplas simulações na cena de *rendering*. A Figura 5.3 descreve os módulos do ambiente.

Os módulos estão divididos em três regiões distintas: entrada de dados, processamento/mapeamento e saída/interação, baseado nas arquiteturas gráficas descritas na Seção 3.1. Na Entrada de Dados (*Data Reader*) são lidos os arquivos no formato *PDB* e, caso necessário, arquivos no formato *OUT*.

Outra forma de interação com o usuário é através da interface gráfica de usuário (*GUI*, ou *Graphical User Interface*) e dos controles de animação (*Simulation Controls*).

A segunda região corresponde ao processamento dos dados, montagem de topologias de aminoácidos (*Topology Builder*), enriquecimentos de informação (*Enhancer* e *Information Enhancer*) e mapeamentos (*Mapper* e *Color Mapper*). Na última, os dados são apresentados graficamente (*Information Manager* e *Renderer*).

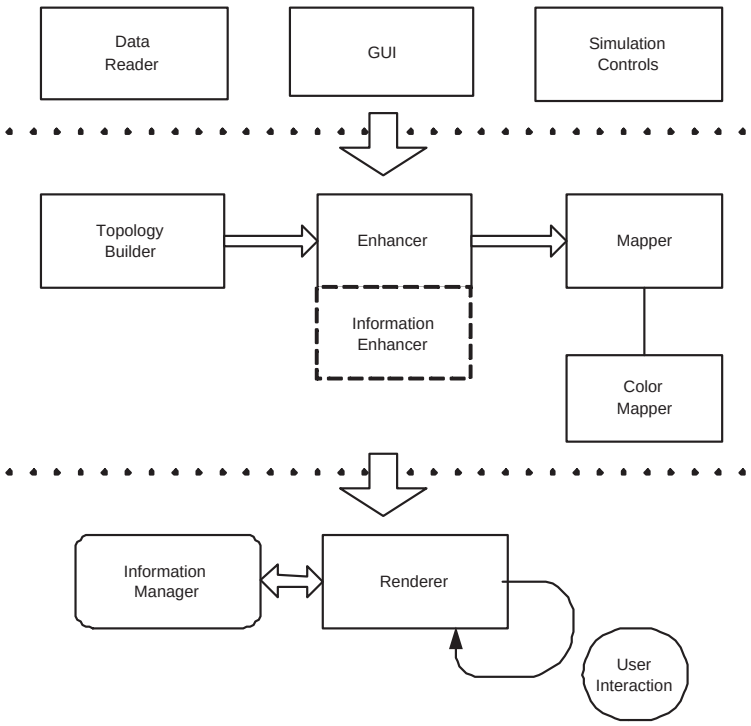

Figura 5.3: Conjunto de Módulos do Ambiente *SimVIZ*.

Estes elementos colaboram para representar informações relevantes na cena de *rendering*. A seguir, descreveremos cada módulo com maiores detalhes, ressaltando suas responsabilidades e características.

5.2.1 Entrada de Dados, Interface, Controles de Animação e Montador de Topologia

O Módulo de Entrada de Dados é o responsável pela abertura dos arquivos de coordenadas atômicas (disponíveis no formato *PDB*), parâmetros utilizados para a execução do *AMBER* e saídas de simulação (arquivos no formato *OUT*) contendo os dados para análise.

Através de janelas, do teclado ou de botões, os usuários informam o que desejam visualizar ou retirar da cena tridimensional, além de controlar a animação da simulação parando, executando ou indo para um determinado tempo. As formas de interação com a interface de usuário e o Manual do Usuário estão descritos no Anexo A.

O Módulo Montador de Topologia conecta os átomos de uma molécula entre si de acordo com os aminoácidos lidos pelo Módulo de Entrada de Dados. Para fazer essas conexões, este módulo baseia-se em informações sobre a topologia dos aminoácidos, que são conhecidas. Caso

o módulo determine que faltam ou sobram átomos, o ambiente notifica ao usuário a existência de problemas com os arquivos lidos.

5.2.2 Enriquecimento de Informações

Este módulo é responsável pelo aumento das informações prévias, tais como elementos estruturais secundários, valores máximos e mínimos dos parâmetros de entrada e saídas e contagens, vistos mais detalhadamente a seguir.

Para realizar essa tarefa, o sistema utiliza duas formas básicas: automática e manual. Na forma automática, o sistema usa os dados de entrada para calcular e preencher as estruturas de dados internas da arquitetura. Na forma manual, o usuário enriquece o sistema (conforme descrito na Seção 3.3.1) com informações que considera relevantes, definindo as informações globais (fixas ao mundo), as relativas ao tempo de simulação (fixa a cena de desenho) ou a elementos gráficos da cena, como aminoácidos (fixa ao objeto).

Para o cálculo e associação a cada aminoácido dos elementos estruturais secundários, foi integrado o *software STRIDE* (ver Seção 2.3) que, além de calcular essas estruturas, verifica os ângulos ϕ e ψ dos aminoácidos. Essa informação é salva nas estruturas de dados do sistema para a representação do Gráfico de Ramachandran.

O Mapa de Contatos necessita dos cálculos das distâncias entre os aminoácidos, todos entre todos. Para a descoberta destes valores, calcula-se o centro geométrico de cada dois aminoácidos e determina a distância entre estes dois pontos, salvando em estruturas de dados específicas (a Seção 2.7.3 explica como este mapa é gerado).

Uma outra forma de enriquecimento de informações utilizado é o cálculo dos valores máximos e mínimos dos parâmetros de entrada e saídas. Esses cálculos são necessários para a construção dos gráficos bidimensionais que são desenhados na cena de *rendering*.

Depois da leitura dos dados, são calculadas contagens para determinar o número de átomos de cada elemento químico constituinte da proteína e o número de aminoácidos de cada tipo. Essas informações são mostradas na cena de desenho.

O ambiente associa arquivos contendo valores de *RMSD* que foram previamente gerados pelo *ptraj*. Esses arquivos são abertos e então são criadas estruturas de dados auxiliares para salvar essas informações. A apresentação posterior é exibida na forma de um gráfico bidimensional (onde o ambiente desenha uma cor diferente para cada arquivo aberto, incluindo um rótulo que contém o nome do arquivo, na mesma cor, para associação visual, conforme a Figura 5.6).

Outra forma de inserção manual de informações que enriquecem a cena é através da abertura da Estrutura de Referência (maiores detalhes na Seção 2.5). Essa estrutura serve para comparação com a estrutura simulada, para que o usuário determine se a simulação está convergindo para um resultado esperado.

As informações associadas aos elementos individuais, passos de simulação ou globais (as *Informações da Sessão*, definidas na Seção 5.1) podem ser consideradas como inseridas no Módulo de Enriquecimento de Informações, mas estão descritas na Seção 5.2.4, que descreve o funcionamento da gerência de informações dentro da cena de desenho.

5.2.3 Mapeamentos

Este módulo constrói os mapeamentos de representações e cores nos átomos e informações adicionais. Para cada átomo são criadas estruturas de dados específicas que serão usadas no Módulo *Renderer*. Cada uma destas estruturas é gerada de acordo com as escolhas do usuário, tanto em termos de representações quanto de raios e definições de elementos gráficos (parâmetros *Stacks* e *Slices* de *OpenGL*, usados para a definição horizontal e vertical dos objetos).

Quando se inicia o ambiente, este verifica se existe o arquivo com as *informações da sessão*. Se o arquivo não existir, ele é criado. Esse arquivo contém os dados que os usuários associaram à simulação (informações globais), aos passos de simulação e aos aminoácidos. Caso o usuário deseje, pode alterar, excluir ou associar novas informações a estes elementos (o Anexo A demonstra este comportamento, na Seção A.4).

Cabe ressaltar que essas informações correspondem à informações abstratas que são inseridas pelos usuários e visualizadas na cena de desenho, de acordo com as definições de *IRVE's* da Seção 3.3. Para o seu desenho serão criados painéis informativos, normalmente transparentes (esses painéis são melhores descritos na Seção 5.3, que trata sobre *IRVE's* e representação de informações abstratas na cena de desenho).

A seguir, vamos explicar os detalhes do Gerenciador de Informações e os elementos gráficos e textuais que são construídos para compor o ambiente.

5.2.4 Gerenciador de Informações

Este módulo lida com a apresentação das informações, onde o usuário determina quais deseja exibir e quais retirar da cena de desenho. Uma das principais atribuições deste módulo é ampliar as informações prévias com enriquecimentos de cena e mapeamentos aos usuários, ou seja, mos-

trar ou ocultar os parâmetros de entrada do *AMBER*, o Gráfico de Ramachandran, o Mapa de Contatos, diversos outros gráficos, as informações abstratas, as visualizações multidimensionais (no caso as Coordenadas Paralelas) e as contagens sobre os átomos e aminoácidos.

O Gráfico de Ramachandran, como explicado no Capítulo 2, representa os ângulos ϕ e ψ dos aminoácidos da proteína. Estes dados foram extraídos do *software STRIDE*, no Módulo de Leitura de Dados (Seção 5.2.1) e salvos em estruturas de dados internas. O Ambiente *SimVIZ* constrói os eixos do gráfico e mostra as regiões proibidas e permitidas aos aminoácidos. Por fim, desenha um elemento gráfico para o par (ϕ, ψ) de cada aminoácido (no caso, um "x", onde sua cor acompanha a forma de colorir utilizada para a proteína).

O Gráfico de Ramachandran da proteína do passo de simulação está representado pela Figura 5.4a. Caso o usuário acrescente a Estrutura de Referência ao ambiente (explicado na Seção 5.2.2), o sistema constrói e desenha o seu diagrama, como mostra a Figura 5.4b. O gráfico desta estrutura é estático, ou seja, não muda ao longo da trajetória, serve apenas como base de comparação. Para a execução da simulação, o ambiente representa um novo gráfico para cada tempo de simulação, ou seja, é dinâmico.

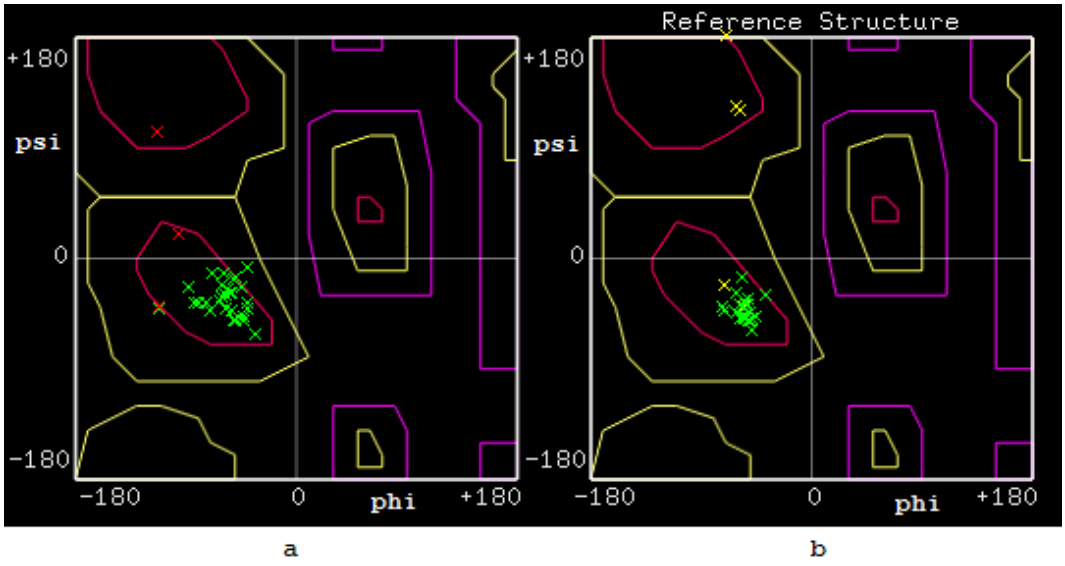

Figura 5.4: Gráfico de Ramachandran. (a) Proteína no passo de simulação. (b) Estrutura de Referência.

A seguir vamos descrever como construímos o Mapa de Contatos para as proteínas. A Seção 2.7.3 explica estes mapas e a sua utilidade no contexto da análise de simulações de trajetórias. Como no Gráfico de Ramachandran, o Mapa de Contatos é desenhado tanto para a proteína do

passo corrente de simulação, quanto para a proteína da Estrutura de Referência, caso tenha sido carregada pelo usuário. A Figura 5.5a mostra o Mapa de Contatos para a simulação e a Figura 5.5b mostra o mapa da Estrutura de Referência.

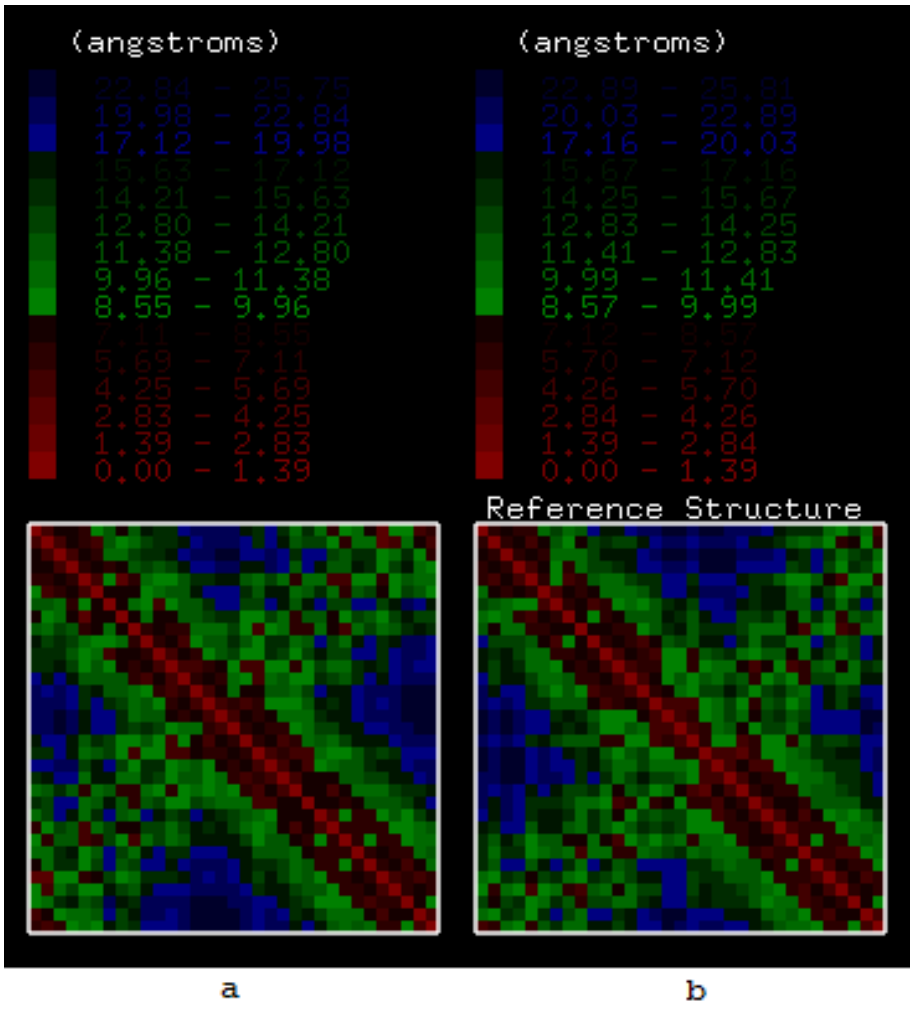

Figura 5.5: Mapa de Contatos. (a) Proteína no passo de simulação. (b) Estrutura de Referência.

Caso o usuário tenha associado arquivos contendo dados de *RMSD*, o ambiente cria este gráfico na cena tridimensional de desenho, colocando o nome do arquivo como rótulo e definindo uma cor padrão. A Figura 5.6 mostra o Gráfico de *RMSD* com quatro arquivos abertos (estes arquivos foram descritos na Seção 5.1).

O *SimVIZ* implementa uma técnica de visualização multidimensional a partir dos atributos de saída de simulações usando Coordenadas Paralelas (ver Seção 3.1.1). Para cada tempo de simulação é criada uma linha que conecta o valor correspondente do atributo naquele tempo, formando as ditas coordenadas paralelas. Como o método é incremental, ou seja, a linha só é

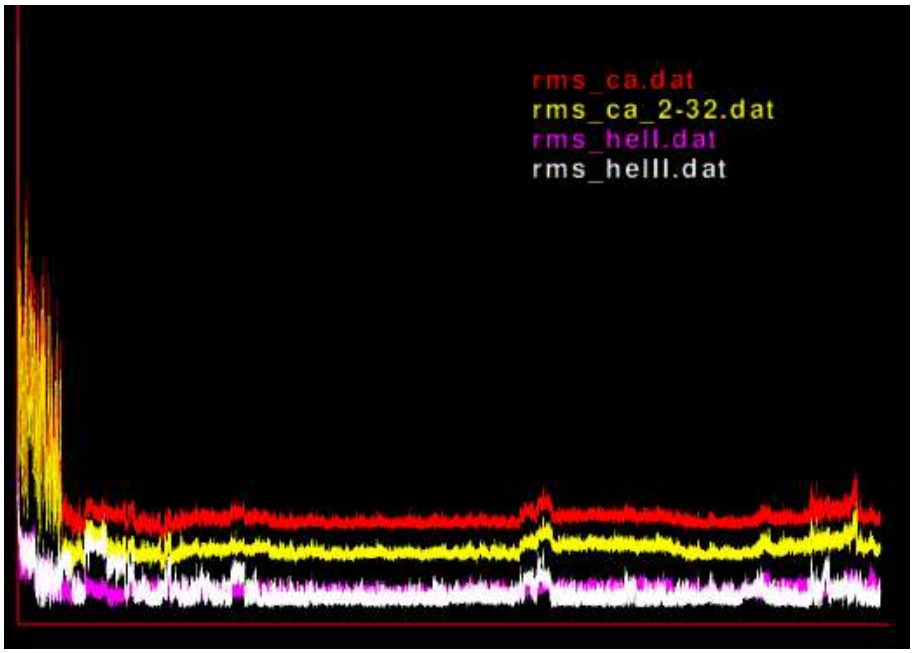

Figura 5.6: Gráfico do *RMSD* de quatro diferentes arquivos de análise.

criada à medida que a simulação é executada, no primeiro tempo de simulação só existe uma linha. À medida que a trajetória é animada (percorre os tempos de simulação, do primeiro passo até o último), novas linhas são criadas. No último tempo de simulação, temos a visualização multidimensional de Coordenadas Paralelas na sua forma final, como mostram as Figuras 5.7 e 5.8. O sistema sorteia uma cor para cada linha criada e os atributos correspondem às saídas das simulações.

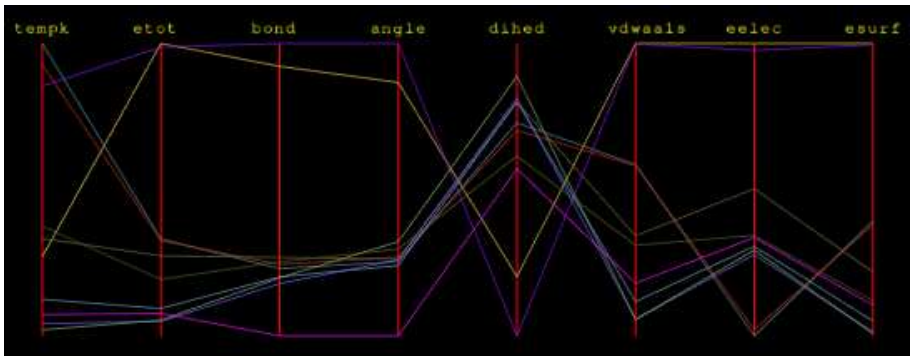

Figura 5.7: Visualização multidimensional de trajetória com Coordenadas Paralelas no décimo nanossegundo do tempo de simulação.

Além das Coordenadas Paralelas, o sistema gera gráficos bidimensionais com os atributos de saída de simulações. Os gráficos são gerados em função do tempo de simulação no eixo x e

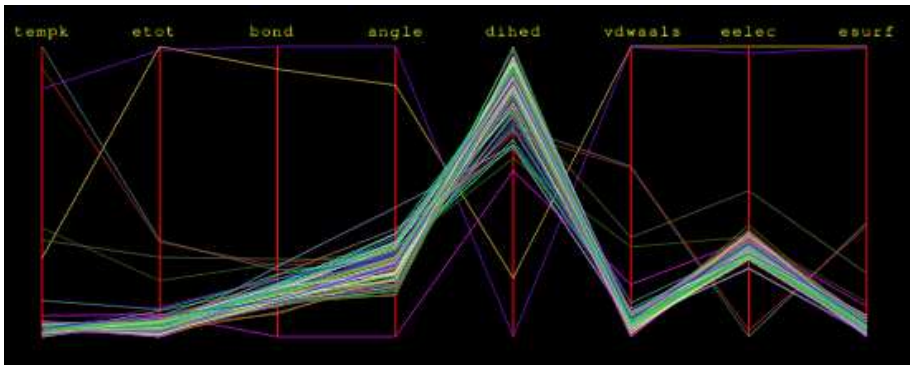

Figura 5.8: Visualização multidimensional de trajetória com Coordenadas Paralelas no último tempo de simulação (centésimo).

sempre correspondem a um atributo da saída no eixo y. Para complementar os gráficos, o ambiente mostra os valores mínimos, máximos, desvios e médias dos atributos em relação à trajetória completa, bem como o valor do atributo no tempo corrente da simulação (ou seja, é uma informação dinâmica, que varia à medida que a simulação executa). A Figura 5.9 mostra os gráficos propriamente ditos e a Figura 5.10 mostra os gráficos com estas informações complementares.

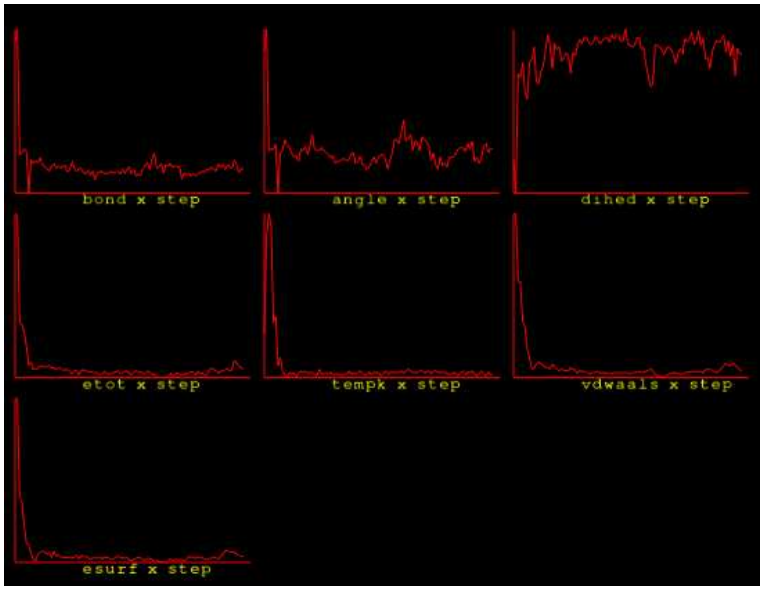

Figura 5.9: Gráficos bidimensionais relativos às saídas de simulação.

O *SimVIZ* mostra as informações sobre as contagens na cena de desenho, uma ao lado da outra, conforme a Figura 5.11. A Figura 5.11a mostra uma relação entre os elementos químicos (C - Carbono, H - Hidrogênio, O - Oxigênio, N - Nitrogênio, S - Enxofre, P - Fósforo) e a Figura

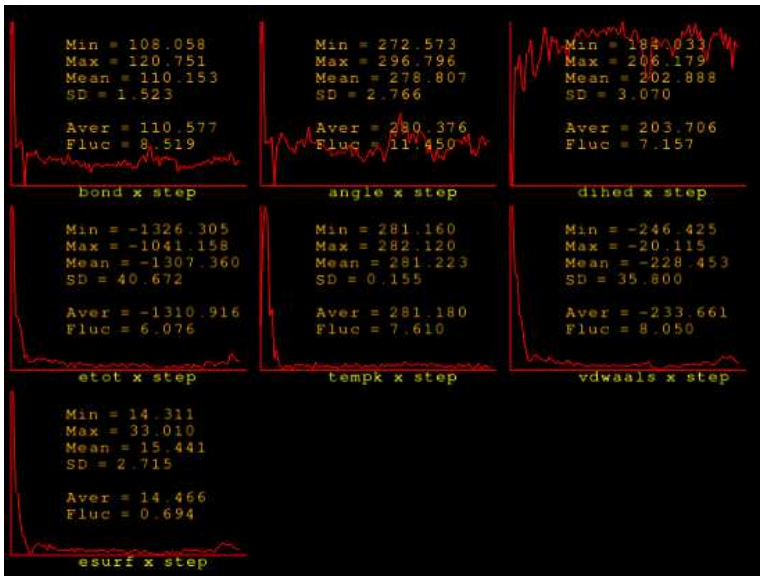

Figura 5.10: Gráficos bidimensionais com as informações complementares.

5.11b mostra a relação entre das quantidades dos aminoácidos usando o código de uma letra (de acordo com a Tabela 2.1). Essas contagens informam uma relação entre o número de elementos existentes e não o valor da contagem propriamente dito.

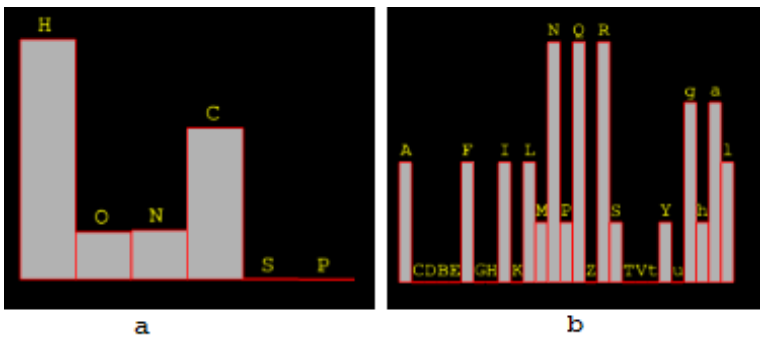

Figura 5.11: Contagens de átomos e aminoácidos. Em a) contagens por elemento químico e em b) contagens por aminoácidos (utilizando o código de uma letra).

Estão disponíveis na ferramenta painéis informativos (ver Seção 3.3.2 sobre os aspectos teóricos envolvidos e Seção 5.3.3 sobre como estamos criando estes painéis no *SimVIZ*) contendo os parâmetros de entrada utilizados no *AMBER*, a saída das simulações e a lista de aminoácidos mais a estrutura secundária calculada pelo *software STRIDE*. A Figura 5.12 mostra esses painéis, construídos com transparências para que o usuário possa ver outros elementos gráficos da cena.

Os painéis foram criados na borda inferior e superior e do lado direito da janela de visuali-

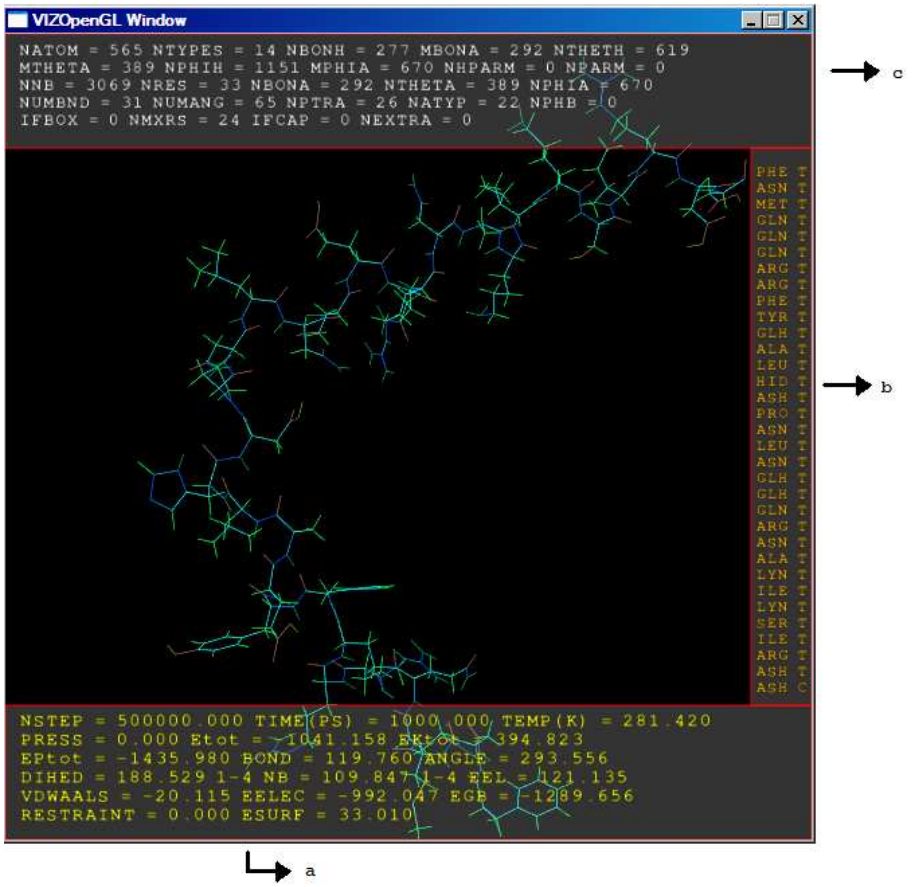

Figura 5.12: Painéis informativos contendo (a) dados de saída do *AMBER* (b) listagem dos aminoácidos e estrutura secundária e (c) parâmetros geométricos da proteína.

zação. O centro foi reservado para visualização de outras informações, gráficos e para mostrar a representação da proteína ou simulação. Criamos os painéis transparentes, para que os outros dados não fossem perdidos ou ficassem oclusos. Estes painéis estão descritos na Seção 5.3.3.

5.2.5 *Rendering*

Após ler e processar os dados, é necessário exibi-los na cena. Este é o módulo responsável por esta tarefa, a partir das informações de mapeamento que executaram anteriormente. Este é um módulo importante do sistema, pois trata da exibição das informações que serão apresentadas escolhidas no Módulo *Information Manager*. Este módulo também faz a interação do usuário com a ferramenta (rotacionando, transladando e escalando) e todas as atualizações gráficas que se façam necessárias.

Construímos as seguintes formas de representações gráficas de proteínas: *Lines*, *VDW*, *CPK*

e *Bonds* (ver Seção 2.6). As formas de colorir disponíveis são: por elemento químico, por nome do aminoácido, por tipo do aminoácido, por cadeia e pelo *backbone*. Desenvolvemos um filtro pelo *backbone* da proteína (este filtro remove os átomos da cadeia lateral dos aminoácidos).

A seguir apresentaremos os tipos básicos de representação de proteínas agregados no ambiente, começando pelo formato *Lines* (conforme Figura 5.13). Por ser uma representação simples trata-se da forma mais rápida de visualização. Nesta forma, apenas as ligações moleculares são desenhadas, sem informações sobre os átomos que formam a proteína. A forma de colorir utilizada na figura foi de acordo com o nome do elemento químico.

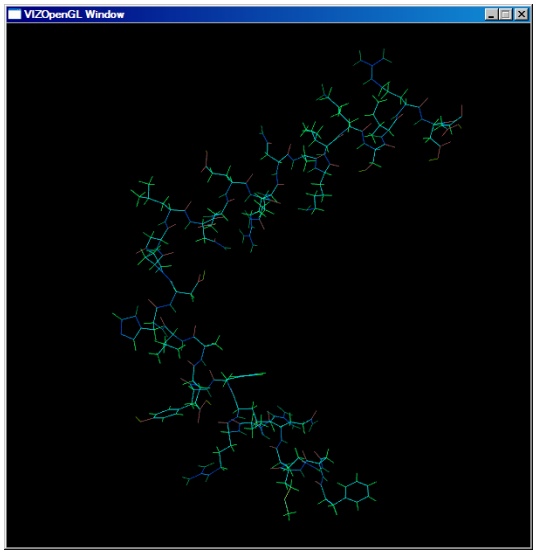

Figura 5.13: Representação no formato *Lines* de uma proteína.

Uma outra forma de representação que construímos foi o formato *Bonds*. Sua única diferença em relação ao formato *Lines* é o fato de não desenhar linhas, mas cilindros. A Figura 5.14 mostra a representação do formato *Bonds*.

CPK desenha os átomos e as ligações químicas entre eles. Os átomos são representados por esferas e as ligações por cilindros. A Figura 5.15 mostra a representação do formato *CPK* de uma proteína. O *SimVIZ* permite a modificação dos atributos do cilindro e da esfera (raio, cor, definição).

A representação no formato *VDW* não desenha as ligações químicas entre os elementos, apenas os átomos, segundo o seu raio de *van Der Waals*. A Figura 5.16 mostra a representação no formato *VDW*. Tal como *CPK*, *VDW* modifica os atributos da esfera que representa o átomo.

Entretanto, desenhar todos os átomos e ligações existentes em uma proteína pode não ser

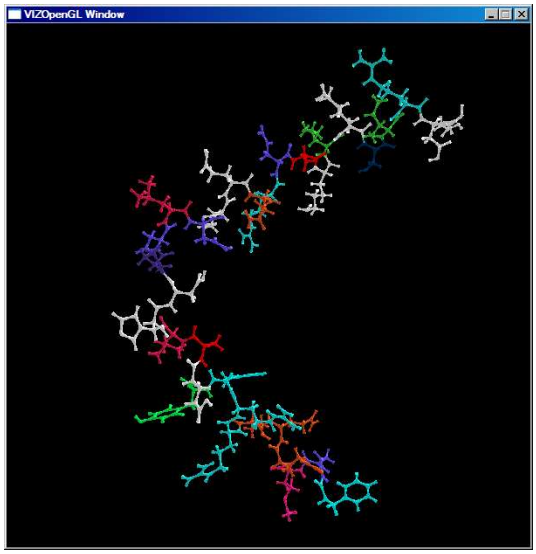

Figura 5.14: Representação no formato *Bonds* de uma proteína colorida de acordo com o nome do aminoácido.

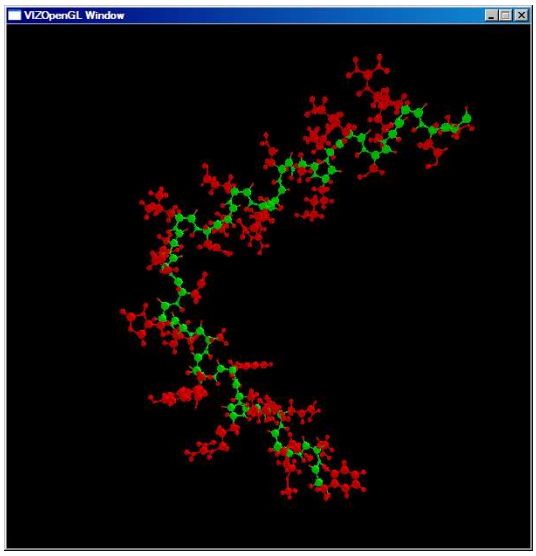

Figura 5.15: Representação no formato *CPK* de uma proteína, colorida pelo *backbone* (a cor verde mostra os átomos do *backbone* e a cor vermelha exibe as cadeias laterais).

a melhor forma de visualização, pois o usuário pode querer analisar apenas o *backbone* e suas flutuações ao longo do tempo. A solução encontrada foi construir um filtro que remove a cadeia lateral do aminoácido, deixando na cena apenas os átomos pertencentes ao *backbone* da proteína. A Figura 5.17 mostra uma proteína onde este filtro foi aplicado. Por retirar elementos gráficos da cena, esse filtro faz com que seja uma visualização mais simples e rápida, em comparação com

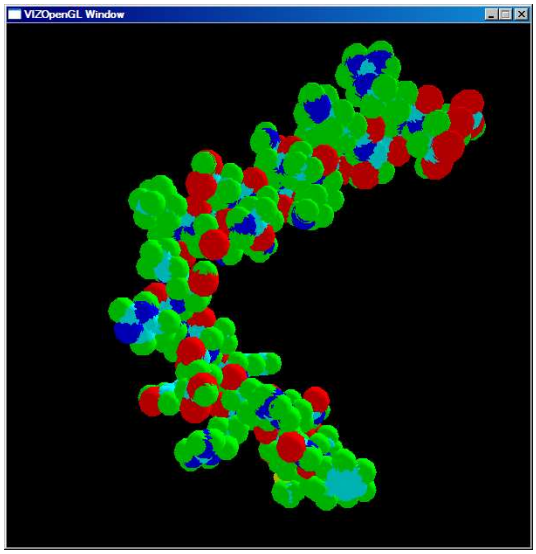

Figura 5.16: Representação no formato *VDW* de uma proteína colorida pelo nome do elemento químico.

a representação da proteína na sua totalidade em termos de elementos.

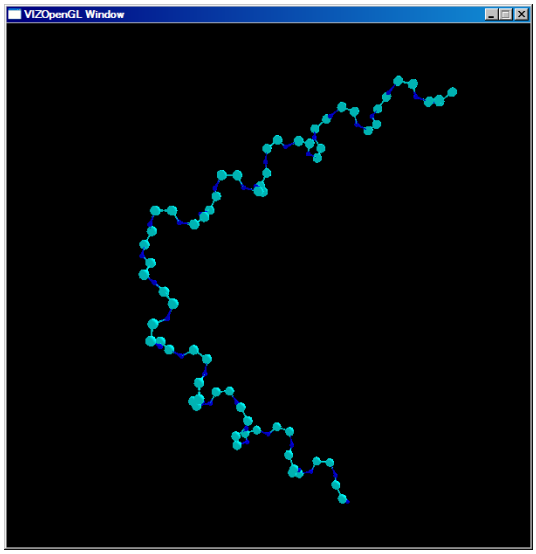

Figura 5.17: *Backbone* de uma proteína representada pelo formato *CPK* e colorido pelo nome do elemento químico, mostrando apenas os átomos de carbono (em azul claro) e nitrogênio (em azul escuro).

Esta seção descreveu os módulos do Ambiente *SimVIZ*. Anteriormente, explicamos as formas utilizadas para ampliar as informações existentes, gerar gráficos e disponibilizar informações de entrada e saída de simulações de trajetórias, individualmente. A próxima seção especifica

as técnicas de *IRVE's* de *Desktop* utilizadas e as decisões de projeto tomadas para permitir a visibilidade das informações na cena de desenho e as associações de informações disponibilizadas aos usuários.

5.3 *SimVIZ* - Ambiente Virtual de *Desktop* Rico em Informação

Os elementos individuais descritos na Seção 5.2.5 são combinados para formar o Ambiente *SimVIZ*, usado para visualização e análise em uma mesma cena de desenho, além de combinações de rótulos, gráficos e visualizações multidimensionais. Esta seção descreve as técnicas de *Desktop IRVE's* utilizadas para a criação do *SimVIZ* e explica as soluções encontradas para a associação e visibilidade de informações, agregação e oclusão e aspectos relevantes que consideramos no projeto do ambiente.

O *SimVIZ* abre e anima trajetórias de simulações de proteínas, ou seja, abre uma série de arquivos que descrevem conformações de proteínas onde cada uma corresponde a um determinado tempo de simulação. A Figura 5.18 mostra 9 tempos de simulação (correspondendo ao tempo de enovelamento da proteína de 1 a 9 nanossegundos). A figura mostra a variação em termos de posição dos átomos em função do tempo de enovelamento e as saídas produzidas pelo *AMBER* e pelo *software STRIDE* (estruturas secundárias).

Detalhando as informações dinâmicas que estão sendo representadas na cena de desenho, vamos recortar a Figura 5.18 separando a lista de aminoácidos (à direita) e estrutura secundária (abaixo) e as saídas de simulação. A Figura 5.19 apresenta a lista dos aminoácidos da proteína e sua estrutura secundária em cada tempo (de 1 a 9 nanossegundos). A figura mostra que a estrutura secundária é dinâmica (no caso, estamos mostrando apenas o código de 1 letra da estrutura secundária, conforme a Tabela 2.2).

A Figura 5.20 mostra as saídas do *AMBER* para os tempos de 1 a 9 nanossegundos da trajetória de simulação. Observa-se que as saídas variam para cada tempo de simulação e que o tempo de 7 nanossegundos possui informações inconsistentes na saída, exemplificando uma característica do ambiente: a inspeção de trajetórias para aumento da qualidade das simulações (esse aspecto é melhor discutido na Seção 5.6).

A visualização da trajetória e a verificação das saídas é um importante aspecto da análise estrutural de proteínas e é usada, por exemplo, para a tomada de decisões quanto à qualidade da simulação. Continuando a descrição do ambiente, explicaremos as representações textuais de

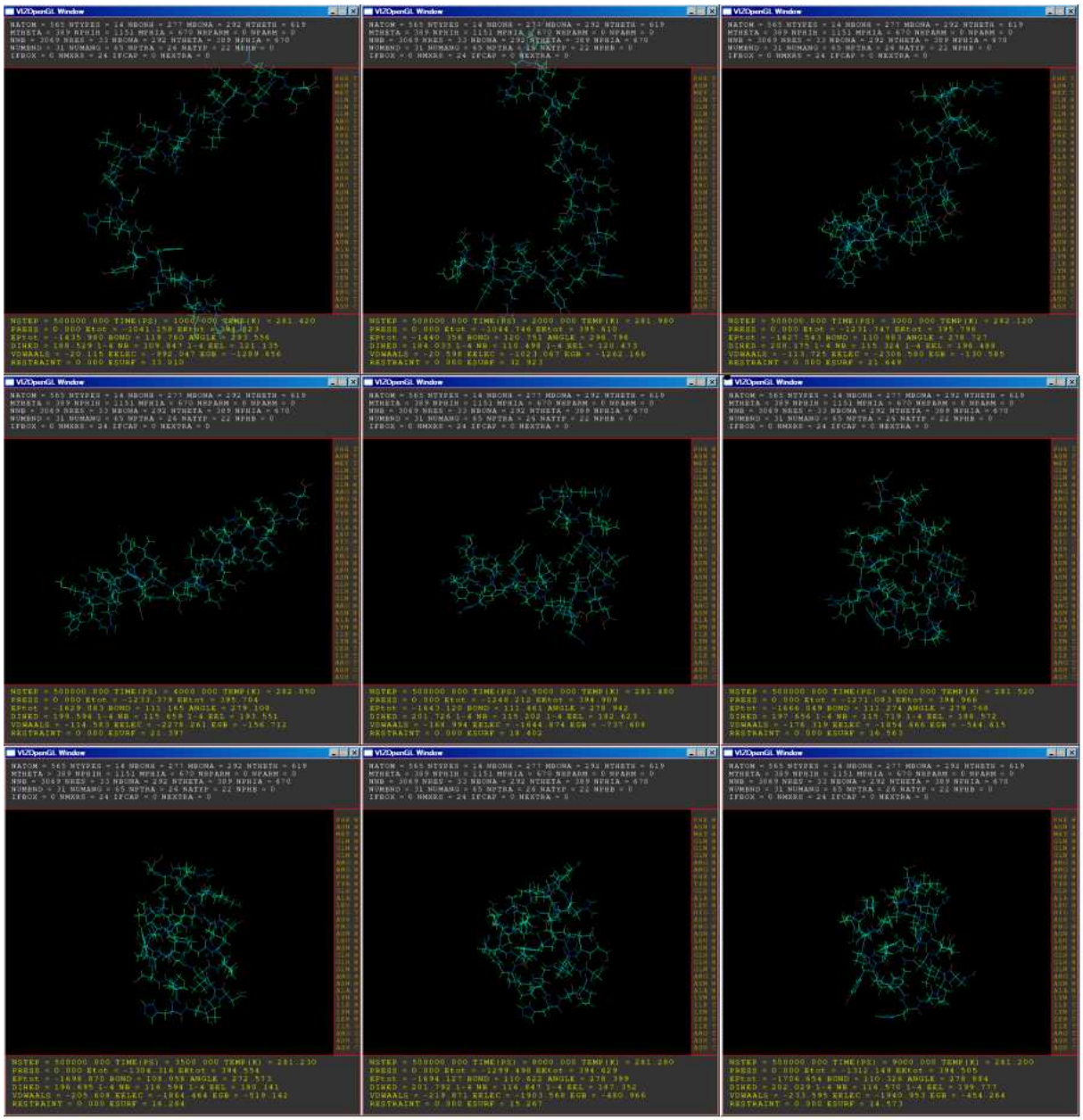

Figura 5.18: Janela de visualização do Ambiente *SimVIZ* ilustrando os tempos de 1 a 9 nanossegundos da trajetória de simulação.

átomos e aminoácidos na cena de desenho.

5.3.1 Representação Textual de Átomos e Aminoácidos na Cena de Desenho

A forma mais simples de representação textual de informações na cena de desenho é a escrita do nome dos átomos e o nome dos aminoácidos. Para escrever o nome (no caso, o código de

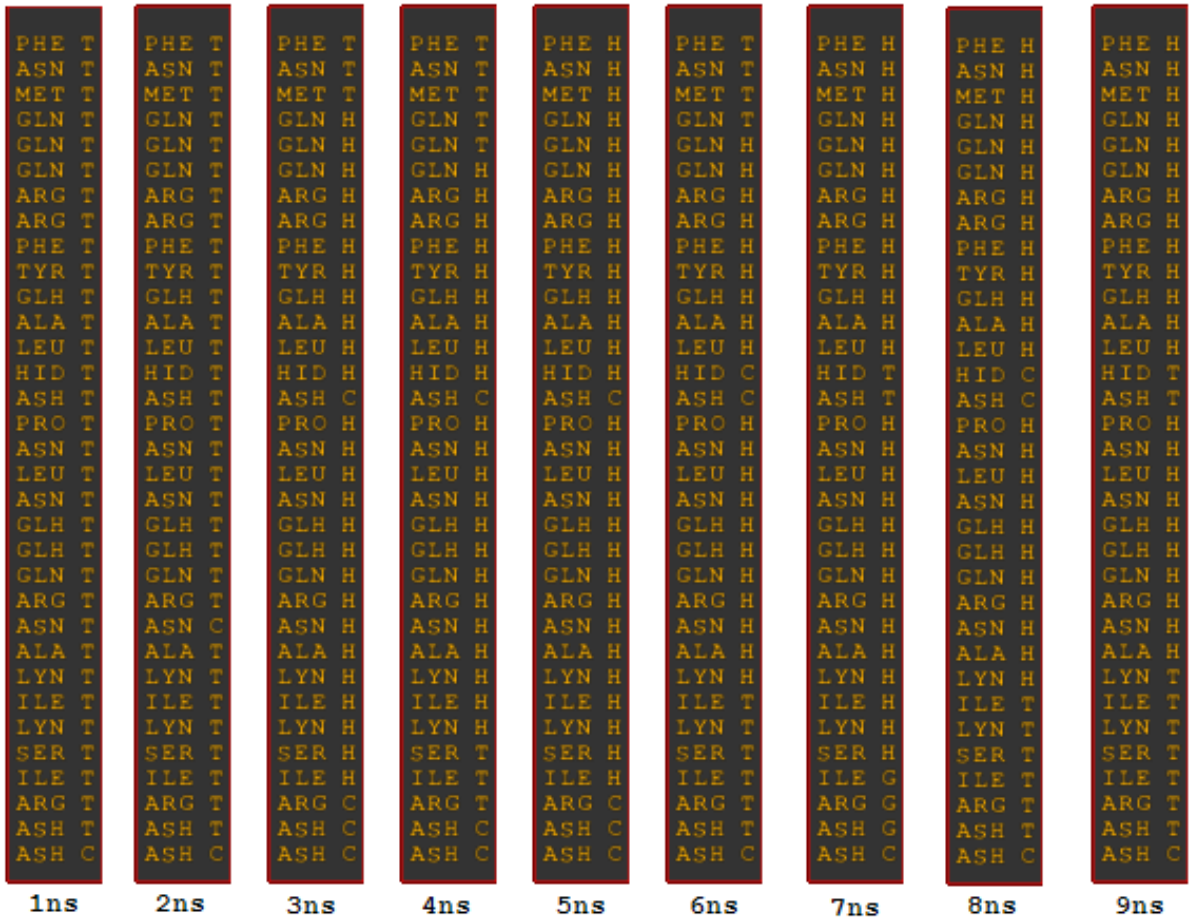

Figura 5.19: Detalhe das estruturas secundárias nos tempos de 1 a 9 nanossegundos da trajetória de simulação.

três letras) dos aminoácidos na cena, o primeiro problema a ser considerado é quanto ao número de aminoácidos existentes. Se esse valor for elevado, tornará a visualização lenta e de difícil entendimento.

Para a visualização textual dos nomes dos aminoácidos, escolhemos usar primitivas gráficas de *OpenGL* para impressão de caracteres. Para posicionar o texto, calculamos o centro geométrico de cada aminoácido a partir da posição dos seus átomos no espaço. A forma de colorir o texto segue a forma de colorir usada na representação.

A Figura 5.21 mostra os nomes dos aminoácidos na cena de desenho. A forma de colorir usada foi pelo nome do aminoácido. Os textos sofrem todas as transformações da representação, ou seja, acompanham o aminoácido mesmo quando a proteína é rotacionada, transladada ou escalonada e sempre apontam perpendicularmente a janela de *rendering*. O tamanho da fonte

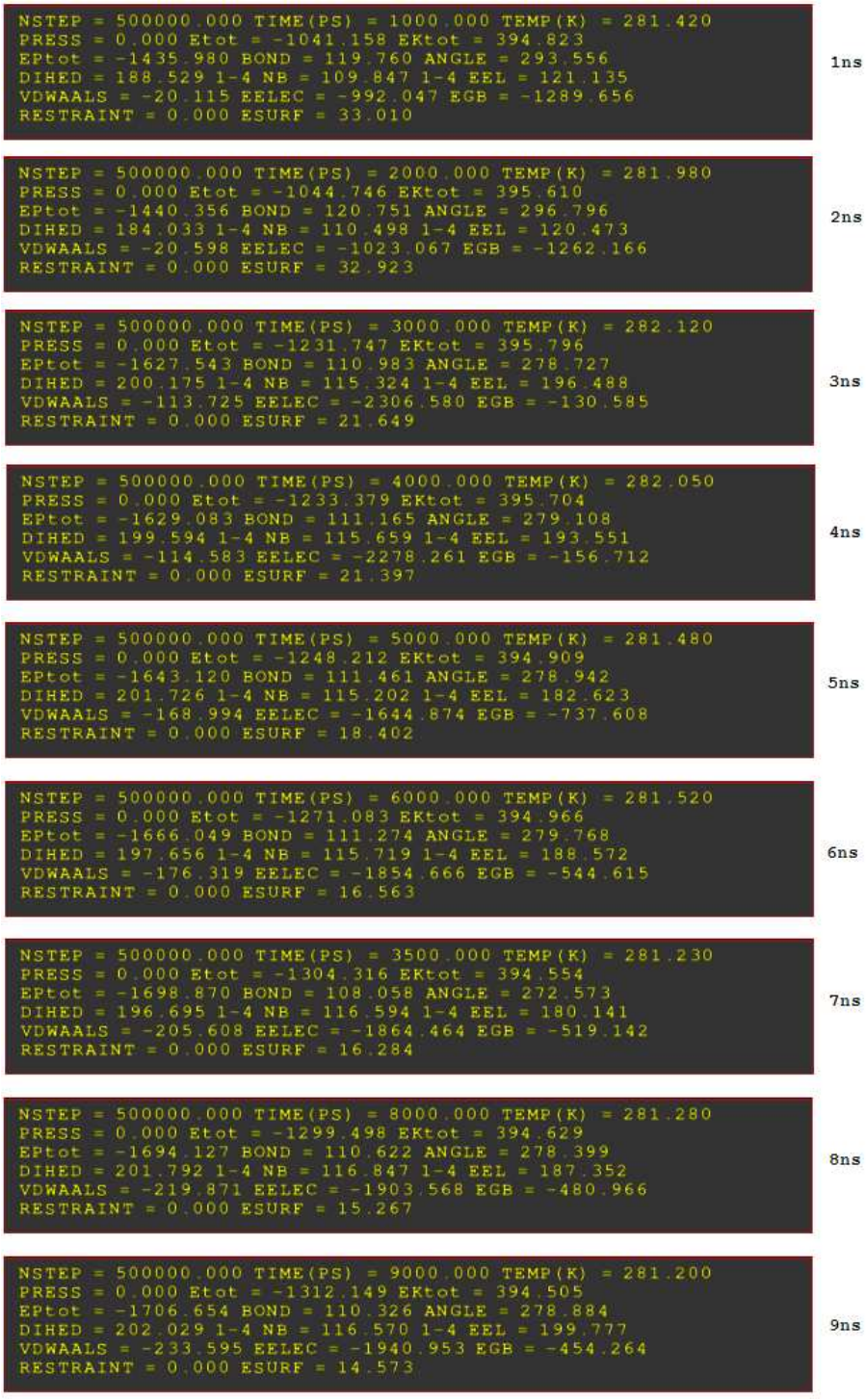

Figura 5.20: Detalhe das saídas de simulação nos tempos de 1 a 9 nanossegundos da trajetória de simulação.

escolhida é definido somente na primeira vez que o objeto é criado, ou seja, não é modificada caso sofra transformações.

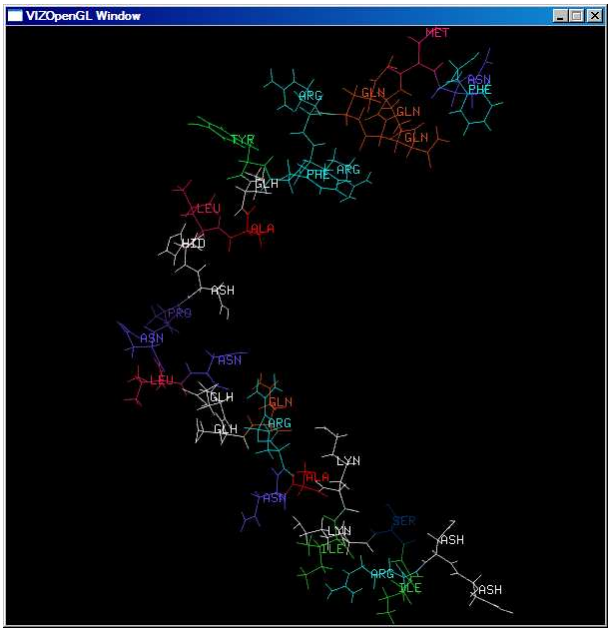

Figura 5.21: Representação textual de todos os aminoácidos de uma proteína.

Implementamos filtros para retirada de determinados nomes de aminoácidos da representação. Na Figura 5.22 estamos mostrando apenas o nome do aminoácido *Alanina* e na representação da proteína, escolhemos colorir de acordo com o nome dos aminoácidos.

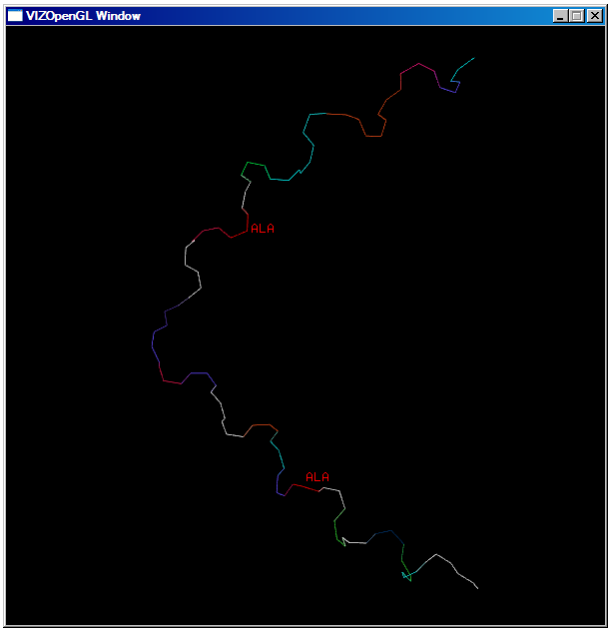

Figura 5.22: Representação textual do nome do aminoácido *Alanina* utilizando o seu código de três letras.

Para o caso dos nomes dos átomos, como seu número normalmente é elevado, mesmo para proteínas consideradas de tamanho pequeno, a solução encontrada foi criar filtros pelo nome do elemento químico. Estes filtros retiram da cena nomes de átomos não desejados. Caso o usuário escolha mostrar todos os elementos químicos, a visualização só é efetiva se combinada com funções de escala (entretanto, o usuário perde informações sobre grande parte da estrutura da proteína, pois não as visualiza). A Figura 5.23 mostra o nome de todos os átomos existentes na proteína. Existem muitas informações para serem vistas. Desenhar e interagir com todos estes elementos gráficos pode comprometer o desempenho e prejudicar a visualização e análise.

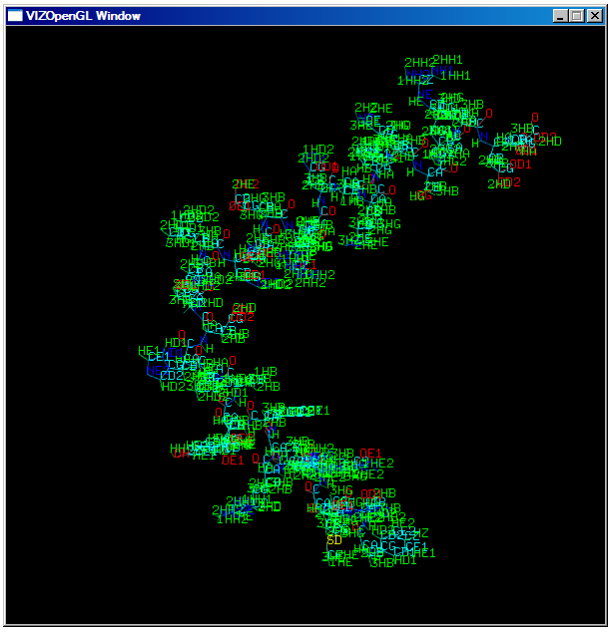

Figura 5.23: Representação textual de todos os átomos de uma proteína sem escala.

A Figura 5.24 demonstra a mesma visualização caso o usuário visualize apenas o *backbone* (*CA - Carbono-α, C - Carbono, N - Nitrogênio*). Há uma melhora em termos de desempenho, pois menos elementos gráficos e textuais foram desenhados.

A Figura 5.25 exibe o *backbone* da proteína com escalonamento. O tamanho do texto não variou, mas ainda está posicionado no mesmo lugar. Observa-se que esta transformação tornou a visualização mais limpa, pois excluiu da cena o restante da representação da proteína.

5.3.2 Localização e Associação de Informações

Esta seção está associada ao módulo de Enriquecimento de Informações, explicado na Seção 5.2.2. Estamos utilizando três formas de localização para divisão da área de desenho sugeridas

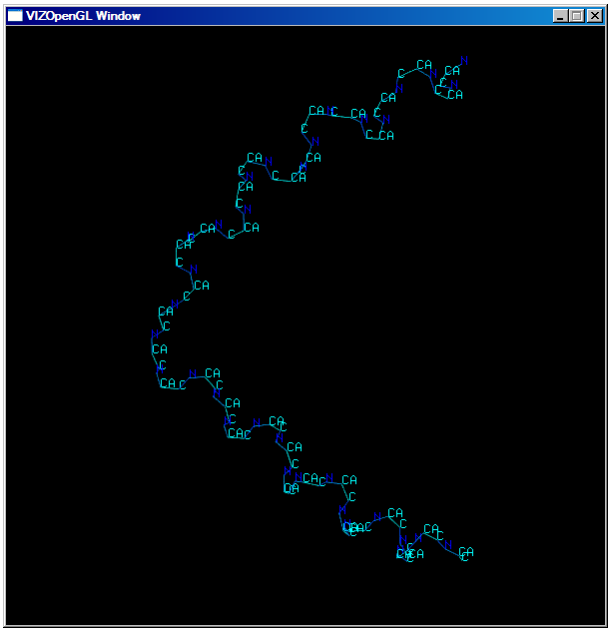

Figura 5.24: Representação textual do nome dos átomos de carbono do *backbone* de uma proteína.

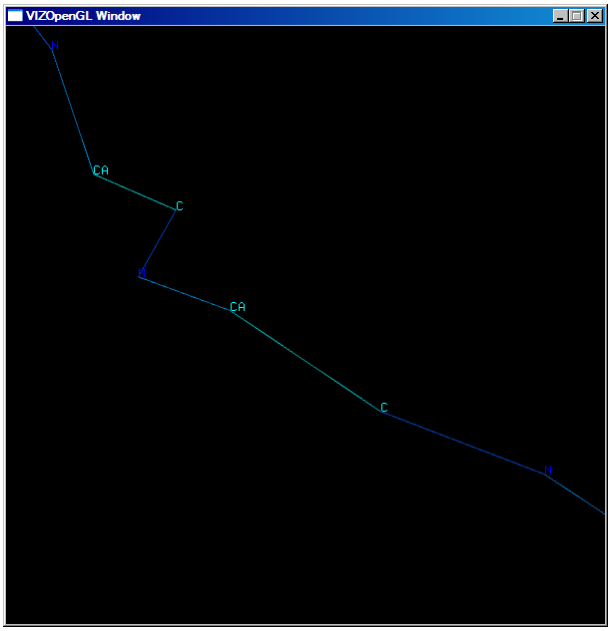

Figura 5.25: Representação textual do nome de todos os átomos do *backbone* de uma proteína com escala.

por *Bowman* (Seção 3.3.2): *fixa ao mundo*, *fixa à cena de desenho* e *fixa ao objeto*. As técnicas foram usadas da seguinte forma:

- Fixa ao mundo: Trata sobre informações gerais da simulação, associadas no ambiente pelos

usuários. Pode ser utilizada para salvar informações tais como data da simulação, em qual agregado (*cluster*) de computadores a simulação foi executada, o autor da simulação e outras informações relevantes;

- Fixa à cena de desenho: Esta forma é utilizada quando informações são associadas especificamente aos tempos de simulação. Em cada passo, o usuário pode constatar determinados eventos dignos de nota, a ponto de desejar associar uma informação. Outras informações tais como gráficos e listas de aminoácidos e estruturas secundárias também são fixas à cena de desenho, e são atualizadas sempre que o tempo de simulação é alterado (ou seja, o ambiente é responsável por estas representações). Essas informações sempre apontam perpendicularmente ao usuário, ou seja, não são tridimensionais (por exemplo, textos deitados ou rotacionados na cena de desenho);

- Fixa ao objeto: Estamos permitindo que sejam associadas informações a aminoácidos específicos de passos de simulação específicos. O usuário pode constatar uma mudança posicional de um aminoácido em um tempo de simulação que seja digna de nota. O ambiente permite que se associe uma informação a este aminoácido (uma anotação).

Uma vez que as informações foram associadas e salvas dentro do ambiente, o próximo passo é representá-las na cena de *rendering*. Escolhemos regiões na cena para desenhar as informações, de acordo com a seu tipo de localização (essas regiões são explicadas na Seção 5.3.3). A *fixa ao mundo* é desenhada sempre, pois é válida para toda a simulação, no canto esquerdo, logo abaixo do *Painel Informativo Superior* (explicado na próxima seção). As *fixas à cena de desenho* são desenhadas a cada passo de simulação, no canto direito, também logo abaixo do *Painel Informativo Superior*. As *fixas aos objetos* são posicionadas perto do centro geométrico dos aminoácidos e desenhadas caso exista uma informação associada. Ressaltamos que essas informações são habilitadas/desabilitadas pelos usuários através de comandos na interface.

5.3.3 Painéis Informativos

Implementamos na ferramenta três painéis informativos principais (mostrados na Figura 5.26): à direita, acima (chamada *Painel Informativo Superior*) e abaixo (*Painel Informativo Inferior*) na cena de desenho. A maior preocupação ao criar esses painéis foi quanto à perda de espaço original na cena de desenho, pois esta ficaria com menos espaço visível. A solução foi tornar os painéis transparentes, assim é possível navegar pelo ambiente sem perdas visuais.

A Figura 5.26 ilustra os painéis informativos criados e sua região em relação à janela tridimensional de desenho. Esta figura mostra a relação entre os painéis e a cena tridimensional de desenho, onde são colocadas as informações dentro do ambiente. À direita é mostrada a lista dos aminoácidos e a estrutura secundária de cada um. No centro da cena, estamos representando as *Informações Globais* (relativas à simulação), as *Informações do Passo de Simulação* (relativas a cada tempo de simulação), os *Gráficos, Mapas e Contagens* e no centro, a *Representação Molecular* da proteína.

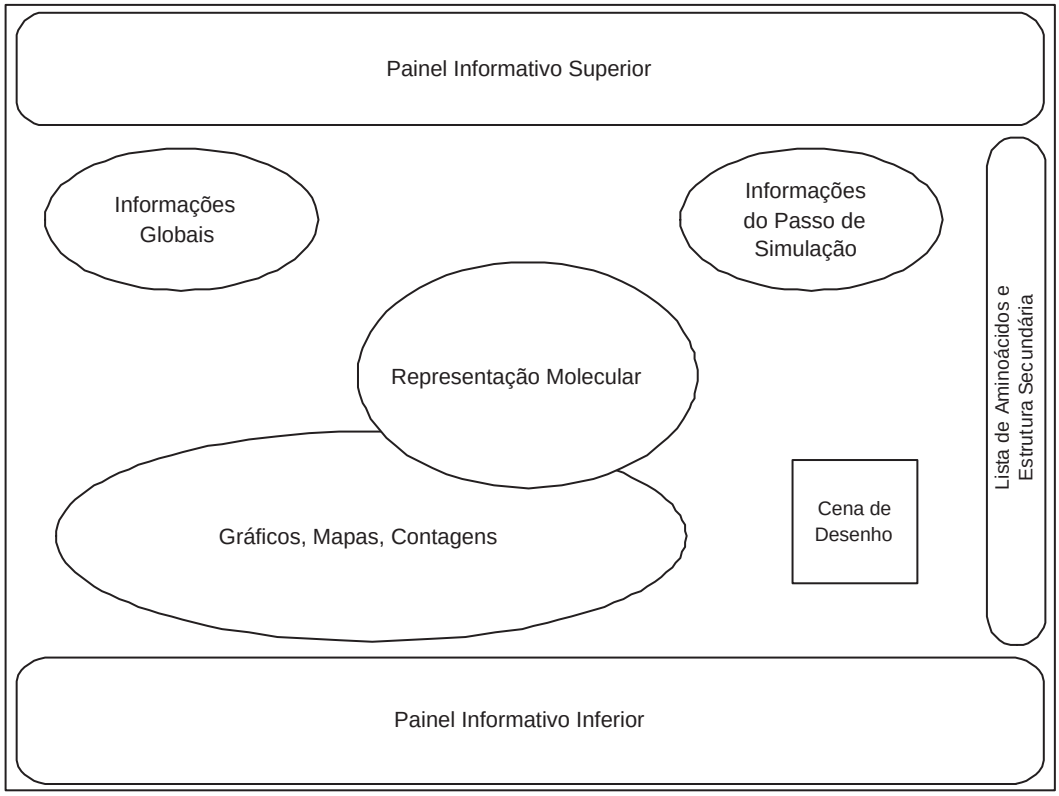

Figura 5.26: Diagrama esquemático dos painéis informativos e regiões informativas do Ambiente *SimVIZ*.

Criamos as regiões na cena de desenho para colocar informações relevantes sobre trajetórias de simulações de proteínas. Escolhemos uma região para colocar os gráficos, mapas e contagens, localizada acima do *Painel Informativo Inferior*. Nessa área estamos desenhando os Mapas de Contatos para a proteína de cada passo individual de simulação e sua Estrutura de Referência, juntamente com o Gráfico de Ramachandran (também, para a conformação da proteína do tempo de simulação e a de Referência) e a visualização das Coordenadas Paralelas.

A região abaixo do *Painel Informativo Superior* foi utilizada para desenhar as informações dos

enriquecimentos de cena restantes. No caso de gráficos bidimensionais de saídas de simulação, dividimos a região central em outras nove regiões, onde cada uma desenha um gráfico diferente.

Descrevemos aqui os painéis informativos do Ambiente *SimVIZ*. A seguir, vamos explicar o processo de integração destes elementos na cena tridimensional de desenho.

5.3.4 Integração dos Elementos Gráficos na Cena de Desenho

A Seção 5.2.4 mostrou os elementos gráficos individuais que implementamos no ambiente. Nessa seção explicaremos como compor a cena com informações variadas e evitar oclusões indesejadas. O problema é a existência de uma quantidade massiva de informações para visualização e análise de trajetórias. O desafio é mostrá-las na janela de visualização de uma forma clara e objetiva.

A seguir, apresentamos as formas encontradas para combinações. Para todas as figuras estamos trabalhando com a representação no formato *Lines* e colorindo pelo nome do elemento químico. A Figura 5.27 mostra a proteína e os três principais painéis informativos. No *Painel Informativo Superior*, são exibidos os parâmetros de entrada utilizados na execução do *AMBER* e à direita a lista de aminoácidos e sua estrutura secundária calculada pelo *software STRIDE*. No *Painel Informativo Inferior*, estão representadas as saídas da simulação.

A Figura 5.28 exemplifica o uso dos gráficos bidimensionais para as saídas de simulação. Neste caso estamos desenhando tanto os eixos quanto os valores na cor vermelha e não estamos representando a proteína, para maior visibilidade. Estamos informando os valores mínimos e máximos para cada atributo, sua média e desvio padrão e as flutuações que existiram nesta execução da simulação (esta informação vem do arquivo *OUT*).

As contagens dos átomos e aminoácidos da proteína do passo de simulação e da Estrutura de Referência são mostradas na Figura 5.29.

O Gráfico de Ramachandran de uma conformação da proteína ao longo da simulação e da Estrutura de Referência está representado na Figura 5.30. O gráfico marca os ângulos ϕ e ψ permitidos aos aminoácidos, colorindo pelo nome do aminoácido.

A Figura 5.31 mostra o Mapa de Contatos da proteína do passo de simulação, à esquerda, e da Estrutura de Referência, à direita. Como mencionado anteriormente, este mapa é dinâmico, sendo recalculado e redesenhado para cada passo de simulação.

Combinando aspectos de representação multidimensional de dados, implementamos a técnica das Coordenadas Paralelas, de acordo com a Figura 5.32. Cada passo de simulação desenha

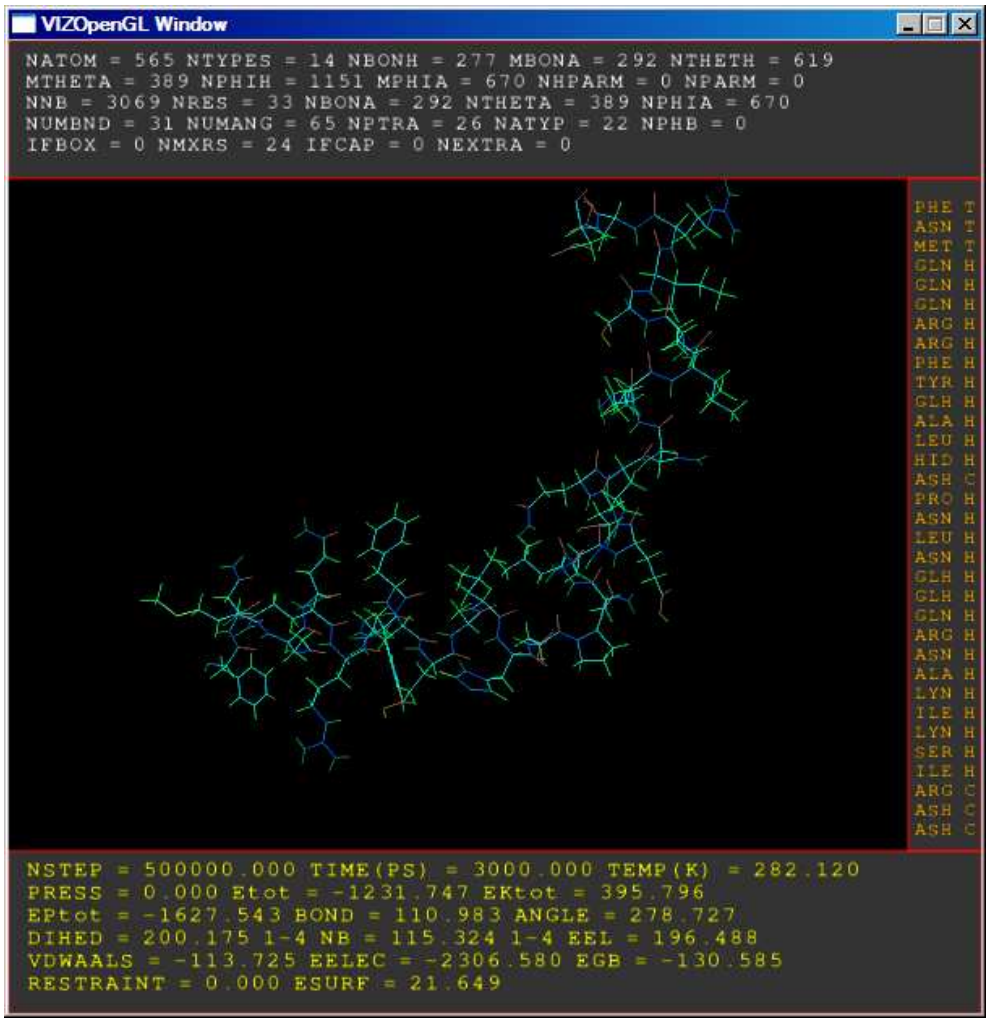

Figura 5.27: Representação de proteína e os painéis informativos.

uma linha que conecta os atributos. Essas dimensões correspondem às saídas de simulação mais utilizadas para análise. Ao fundo, na cena de desenho, estamos combinando a visualização com a representação da proteína.

A Figura 5.33 mostra o gráfico para análise do *RMSD* da estrutura com 4 arquivos de análise gerados pelo *ptraj*.

As informações globais, relativas ao tempo de simulação e relativas aos aminoácidos são representadas pela Figura 5.34, juntamente com a representação da estrutura da proteína e os painéis informativos.

Nesta seção apresentamos as formas de combinação de gráficos, textos, representações de proteínas e técnicas de representações multidimensionais desenvolvidas no Ambiente *SimVIZ*. Aos usuários é permitida interação com atualização automática, ou seja, alteração de raios,

Figura 5.28: Painéis informativos e gráficos bidimensionais.

cores, representações, à medida que a animação da trajetória é executada.

Um problema que detectamos e que fica a critério do usuário é o da oclusão, já que não implementamos um gerenciador de janelas responsável pela descoberta de quais informações colocar em cada região da cena de desenho. O usuário seleciona as informações, gráficos e mapas que deseja ver e estes são desenhados na cena, em uma posição predeterminada.

Esta seção descreveu as formas de integração de gráficos e textos para compor a cena tridimensional de desenho. A seguir, explicaremos a arquitetura do Ambiente *SimVIZ*, com uma breve descrição das classes criadas e suas responsabilidades.

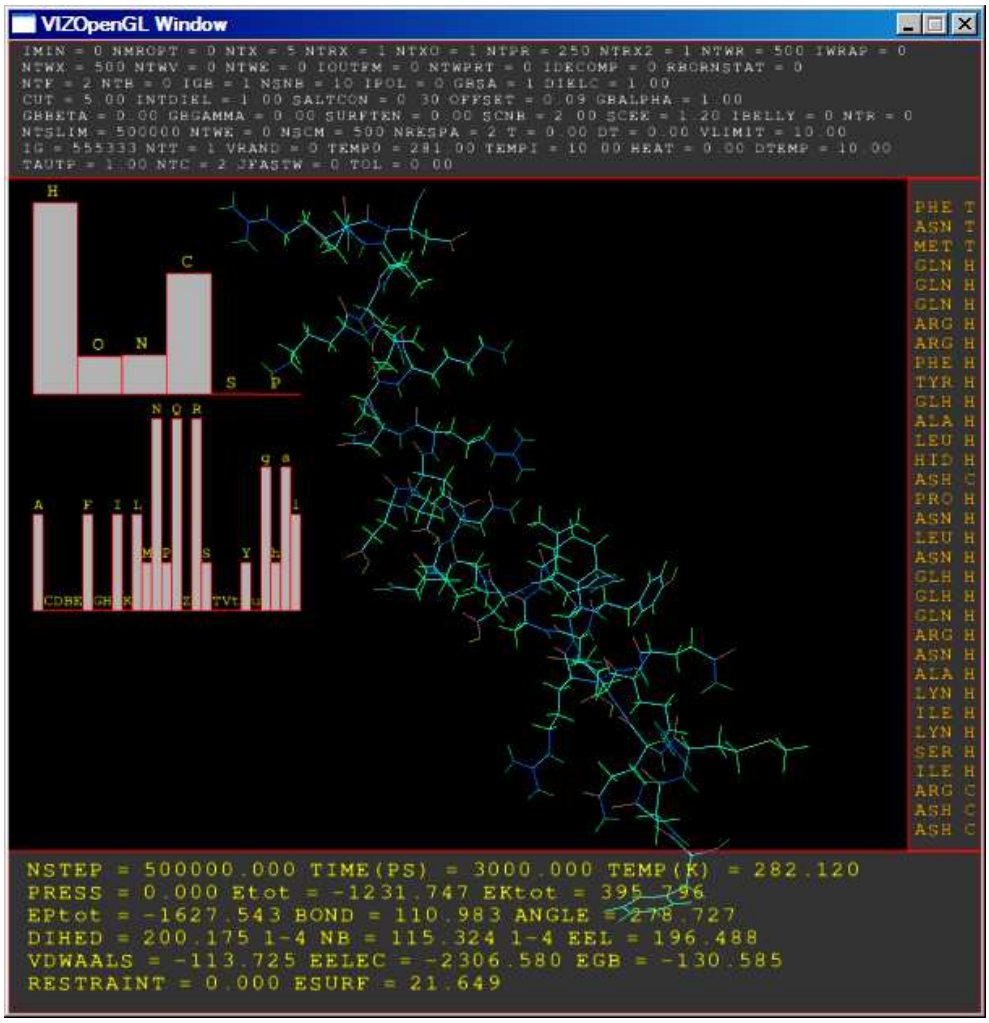

Figura 5.29: Painéis informativos e contagens de átomos e aminoácidos.

5.4 Arquitetura do Ambiente *SimVIZ*

O Diagrama de Classes simplificado na notação *UML* (*Unified Modelling Language*) está ilustrado na Figura 5.35 e mostra o modo como as classes da arquitetura estão associadas entre si. Esta arquitetura é baseada em uma mais simples chamada *VIZ* [23]. A arquitetura do Ambiente *SimVIZ* utilizou e estendeu definições da *VIZ*.

A seguir discutiremos as principais responsabilidades de cada classe:

VIZ: Esta classe é responsável por guardar as estruturas de dados principais da ferramenta. Além de ser global ao escopo da aplicação, também executa o *software STRIDE* internamente, tanto para gerar arquivos temporários quanto para extrair informações de arquivos (realizar um *parsing*) e guardar os elementos das estruturas secundárias (α-hélices, folhas-β, voltas e alças).

Figura 5.30: Representação do números de átomos e do Gráfico de Ramachandran.

Outra importante responsabilidade é controlar a simulação (através dos comandos *play, stop* e *goto*). Esta classe possui uma lista de volumes (um volume representa uma proteína), para a abertura de múltiplas trajetórias;

VIZVolume: Esta classe representa uma proteína, ou seja, uma lista de aminoácidos. Um volume possui uma lista de moléculas do tipo *VIZMol*. As responsabilidades de um volume é guardar a lista de aminoácidos e calcular os valores máximos e mínimos das saídas de simulação, normalizando os dados (através da classe *VIZSimStats*);

VIZMol: É um mapeamento para uma molécula. Possui uma lista de aminoácidos e uma lista de átomos (para facilitar o desenho). Possui uma classe *VIZSimInfo* que guarda as informações para esta molécula que correspondem a um tempo de simulação. Esta classe é responsável por acessar as informações produzidas pelo *software STRIDE* e determinar o início e o fim de cada

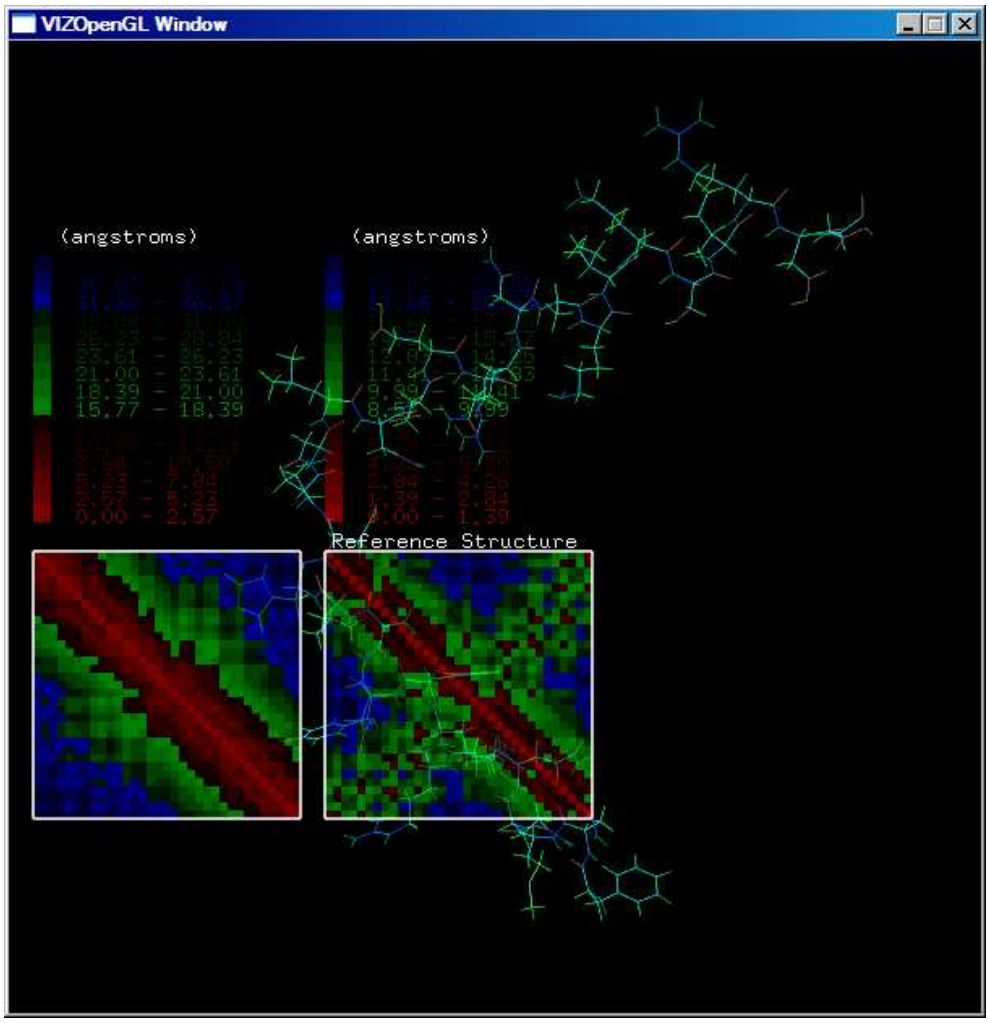

Figura 5.31: Mapa de Contatos da proteína e da Estrutura de Referência.

elemento da estrutura secundária. Isto é importante para a fase de ampliação das informações (a seguir, na Seção 5.2, Módulo *Enhancer/Information Enhancer*);

VIZResidue: Esta classe guarda uma lista de átomos de um determinado aminoácido (ou resíduo), salvando o próximo e o anterior, para permitir que sejam ligados entre si. Esta classe é responsável por criar as topologias de cada resíduo da proteína, preenchendo a sua lista de átomos;

VIZAtom: Representa um átomo na cena tridimensional. Possui diversas informações relacionadas a átomos, como nome, símbolo, suas coordenadas, tipo, seu resíduo atual e uma lista de átomos que este átomo está ligado. Ainda, possui informações sobre como será sua representação e de que forma são as ligações na sua lista de átomos;

VIZGLWindow: Esta é a classe responsável pelo desenho dos objetos na cena tridimensional.

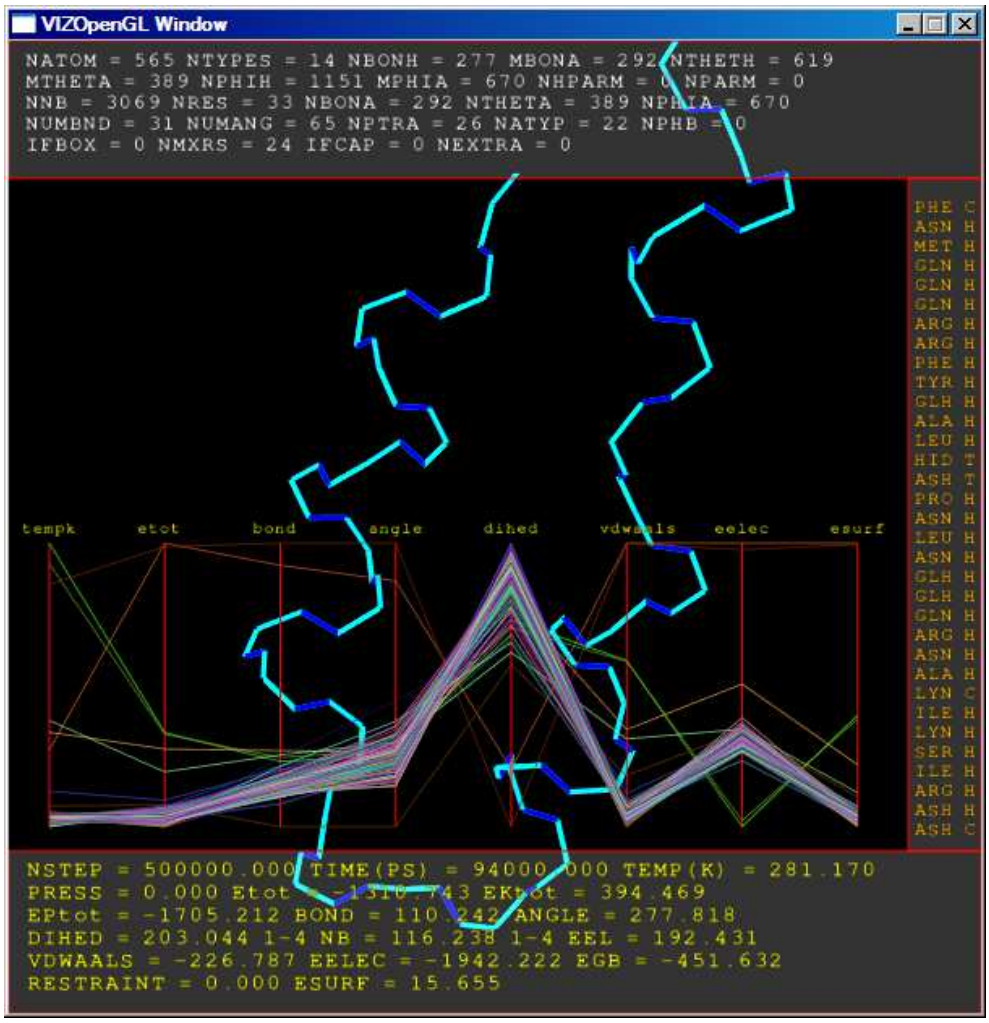

Figura 5.32: Representação de proteína e visualização das Coordenadas Paralelas.

Esta classe ainda é responsável pela criação de representações e mapeamentos para primitivas gráficas *OpenGL*. A *VIZGLWindow* utiliza uma biblioteca gráfica que implementa uma hierarquia para movimentação e posicionamento e *rendering* de estruturas chamada *SmallVR* (entre outras funcionalidades, voltadas a sistemas de Realidade Virtual) [43, 12];

VIZColorer: Esta classe é utilizada para criar uma cor dependendo da forma de cores escolhida pelo usuário;

VIZSimStats: Como explicado em *VIZVolume* esta classe normaliza os dados de saída das simulações. Calcula os valores máximos, mínimos, a média e o desvio para a futura construção das escalas. Esta classe determina o conjunto de valores possíveis para cada dado;

VIZSimInfo: Esta classe representa os dados de entrada e saída de simulações. Outra função importante desempenhada por esta classe é carregar um arquivo de saída de simulação gerado

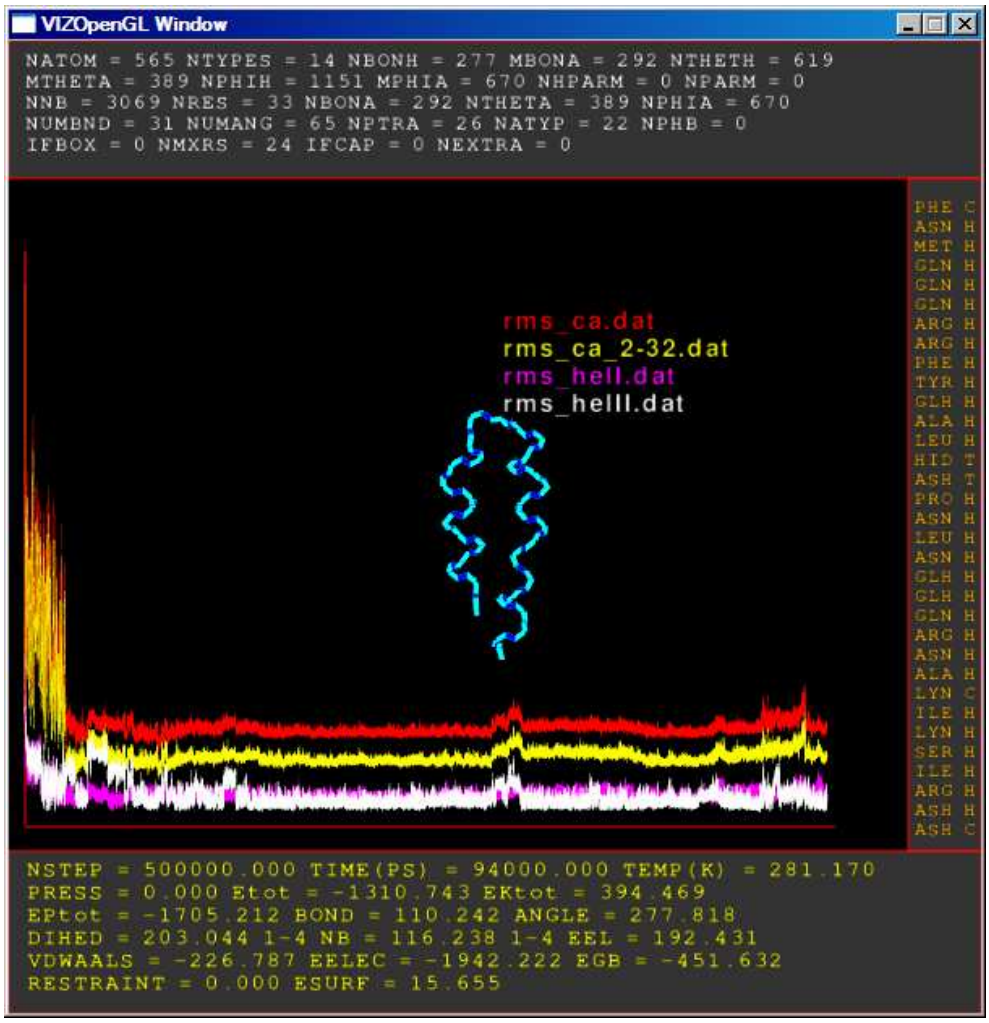

Figura 5.33: Representação de proteína e visualização do gráfico de *RMSD* para 4 arquivos de análise.

pelo *AMBER*;

VIZGLLine e VIZGLCylinder: Guarda informações sobre uma linha, tais como definição, raio inicial e final, comprimento e tamanho. Normalmente se refere a uma ligação entre dois átomos;

VIZGLSphere: Trata-se de informações sobre uma esfera, tais como definição, raio e tamanho. Normalmente se refere a um átomo;

VIZInfoBase: É uma classe base para guardar informações, sendo estendida de *VIZ*, *VIZVolume*, *VIZMol* e *VIZResidue*, já que o sistema permite que sejam associadas para estes elementos (ver Seção 5.3.2);

VIZFont: Trata sobre os aspectos de manipulação e *rendering*, seleção, cores e tamanhos de fontes no sistema;

Figura 5.34: Visualização de informações associadas (a) globais, (b) ao passo de simulação e (c) a dois aminoácidos específicos.

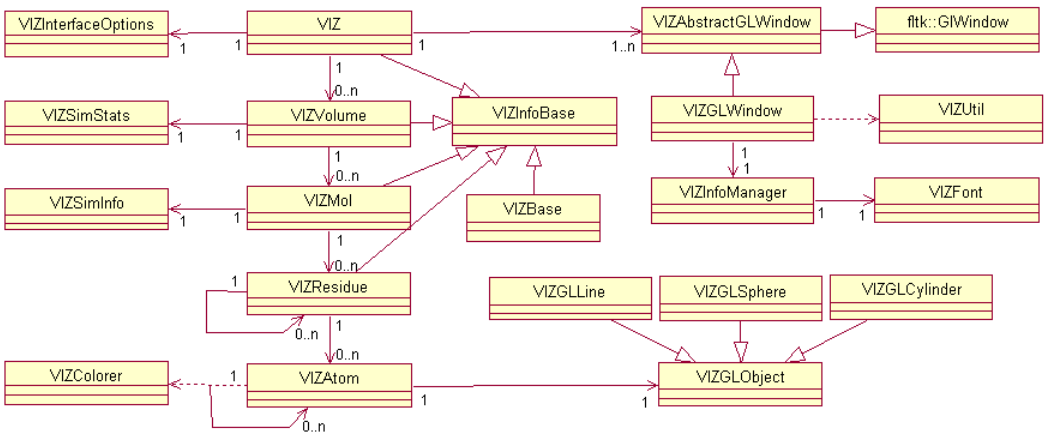

Figura 5.35: Arquitetura do Ambiente *SimVIZ* utilizando a notação simplificada do Diagrama de Classes de UML.

VIZInfoManager: Esta classe lida com todos os aspectos relacionados às informações relevantes na cena de desenho, desde a criação de eixos de gráficos até o *rendering* das fontes. A

VIZInfoManager é o cerne do ambiente rico em informação que implementamos, e realiza todas as complexidades envolvidas na criação das Coordenadas Paralelas, representação do *RMSD*, definição de cores e transparências, rótulos, posicionamento de elementos gráficos, contagens, estatísticas e painéis informativos.

Nesta seção foram apresentadas e explicadas as classes mais importantes do ambiente e suas responsabilidades. A seguir ressaltamos as considerações finais sobre o ambiente.

5.5 *Feedback* de Usuários

Com a finalidade de descobrir informalmente as vantagens e desvantagens do ambiente, bem como detectar sua usabilidade e facilidade de uso, realizamos uma entrevista com usuários. Para tanto, apresentamos o sistema para dois potenciais usuários, que comentaram as partes fracas e fortes do ambiente, bem como suas diferenças principais em relação a outras ferramentas.

Foi levantado que realizar as tarefas em um mesmo ambiente é uma funcionalidade útil, por exemplo, a abertura dos arquivos de RMSD. Antes era necessário executar uma ferramenta externa, implicando em gastos de tempo desnecessários. Sobre o aspecto da ferramenta não disponibilizar representações gráficas como o formato *Ribbons*, foi constatado que o interesse principal é sobre a trajetória da simulação, sendo o formato *Lines* uma boa forma de representação.

Ao disponibilizar os parâmetros do *AMBER* os usuários identificaram que esta é uma funcionalidade importante, pois torna verificável a qualidade da simulação por inspeção visual. Esta visualização dos parâmetros permite identificar se o protocolo executou corretamente, para cada tempo de simulação e se a trajetória teve sua saída afetada por este motivo.

Como principais desvantagens, os usuários destacaram que a interatividade poderia ser melhorada, e as funções de rotação, translação e escala poderiam ser melhor tratadas, sendo mais fácil de serem utilizadas. Também discutiram que algumas formas de colorir poderiam ser exploradas e que são usadas em análise, tal como a forma de colorir pelo tipo do aminoácido (hidrofóbico, polar, entre os discutidos na Seção 2.2.1).

Os usuários sugeriram mudanças quanto à posição dos elementos textuais na cena, mas destacaram a importância destes serem coloridos juntamente com a forma de colorir da proteína que esteja sendo utilizada. Outra desvantagem apontada foi o fato de existirem muitas informações na cena de desenho. Mas quando explicado que o sistema abre múltiplas janelas e que é possível ver diferentes informações em diferentes janelas, essa desvantagem foi amenizada, inclusive pelo

fato da ferramenta abrir simulações e proteínas *stand-alone* (fato importante para análise).

5.6 Considerações Finais Sobre o Ambiente *SimVIZ*

Neste capítulo apresentamos o ambiente de *desktop* rico em informação chamado *SimVIZ*, descrevendo suas funcionalidades, características, módulos e arquitetura. O ambiente, por abrir múltiplas trajetórias de simulações de proteínas por DM, permite que sejam visualizados e analisados os aspectos estruturais de proteínas, onde informações relevantes são exibidas na cena de desenho. Este sistema utiliza informações de um variado espectro como gráficos, Coordenadas Paralelas, Mapas de Contatos, informações textuais para criar um ambiente de visualização e análise de trajetórias de simulações de proteínas.

Como mencionado no Capítulo 3, *IRVE's* e *IRVE's* de *Desktop* combinam conceitos de Ambientes Virtuais com Visualização de Informações. O ambiente proposto e desenvolvido é uma tentativa em direção à integração destas duas áreas, oferecendo um local que mostra animações de trajetórias, mudança de representações e interatividade com o usuário. As ferramentas que implementam *IRVE's* não combinam diferentes técnicas de visualização multidimensional, como Coordenadas Paralelas. Este é um dos diferenciais do ambiente construído.

Outro diferencial tange os problemas que relacionamos quanto à oclusão. Mencionamos na Seção 5.3.4 que o problema da oclusão era deixado para o usuário resolver. Porém, ao permitirmos que múltiplas trajetórias sejam abertas em múltiplas janelas de desenho, estamos resolvendo, em parte, este problema. Se abrirmos a mesma simulação duas vezes, o sistema realizará todos os cálculos e o usuário poderá escolher mostrar diferentes gráficos e textos em diferentes janelas. Em uma janela o usuário pode estar vendo a representação e os painéis informativos sobre as saídas e na outra janela de desenho ele estará analisando o Gráfico de Ramachandran e as estatísticas sobre a simulação.

Um outro aspecto verificado foi quanto aos parâmetros de entrada usados pelo *AMBER*. Como são dados estáticos sobre a simulação, o esperado é que eles não variem ao longo do tempo de simulação. No entanto, dependendo do protocolo de simulação que está sendo conduzido, estes parâmetros podem variar para cada bloco de simulação. O Ambiente *SimVIZ* constrói visualizações para que essas alterações entre os passos sejam visíveis, para análises posteriores.

Como foi verificado ao longo do capítulo, o *SimVIZ* pode ser usado para inspecionar trajetórias de simulações em busca de problemas e erros ocasionados na fase de produção de resultados

de análise. Esta funcionalidade é importante pois determina a qualidade da trajetória e é útil para a tomada de decisão. Um sistema que proporcione essa verificação de erros é fundamental para que visualizações e análises sejam conduzidas da melhor forma possível, fazendo com que os usuários adquiram confiança sobre os dados produzidos pelos *softwares* externos ao *SimVIZ*.

Este capítulo descreveu o Ambiente *SimVIZ* e suas características. O próximo capítulo trata sobre as conclusões e os trabalhos futuros.

Capítulo 6

Conclusões

Esta dissertação definiu uma arquitetura e um fluxo de execução para um ambiente informativo para visualização e análise de trajetórias de simulações de proteínas, usando conceitos de Ambientes Virtuais de *Desktop* Ricos em Informação. Estes objetivos foram plenamente alcançados uma vez que detectamos que ambientes deste porte careciam de integração com técnicas novas de visualização de informação, ou seja, ainda existe muito trabalho nesta área de pesquisa em termos de integração de novas formas de ver a informação e extrair padrões e relacionamentos de forma simples e significativa.

Foram estudados conceitos relativos a Bioinformática Estrutural tais como proteínas, aminoácidos, trajetórias de simulações e visualização multidimensional, interação e apresentação de informações relevantes em uma cena de desenho. Notamos a importância de se estudar as conformações das proteínas ao longo do tempo de enovelamento, para verificar e entender este processo de forma mais apropriada. Estes conceitos foram muito importantes no desenvolvimento do trabalho, pois definiram a problemática que estávamos lidando e serviram como base para a construção do sistema.

Tratar com os aspectos inerentes de integração de visualização e análise de proteínas provou ser desafiador, uma vez que trabalhamos com dados multidimensionais e temporais. Outro desafio que constatamos foi sobre a integração de múltiplas bibliotecas em um mesmo sistema, onde cada uma desempenhou um papel importante. Por exemplo, usamos uma biblioteca para desenho, uma para a interface, uma para mostrar informações textuais com diferentes possibilidades de fontes, tamanhos e cores e uma para posicionar os elementos na cena e interação com a estrutura.

Ao término do trabalho, verificamos que cumprimos os objetivos propostos e implementamos um protótipo de um ambiente virtual de *desktop* rico em informação. Ainda existem muitas

possibilidades para estender esse ambiente, lidando com diferentes aspectos de Bioinformática e Ciência da Computação, vistas com maiores detalhes na próxima seção.

6.1 Trabalhos Futuros

Durante o projeto e implementação do ambiente, pensamos em diversas formas de ampliar o conhecimento sobre trajetórias de simulações de proteínas e cada nova idéia está descrita a seguir. Como a Bioinformática Estrutural é um campo multidisciplinar, existem aplicações para áreas distintas da Ciência da Computação, tais como Interação Humano-Computador, Sistemas de Informação e Processamento Paralelo e Distribuído. O ambiente também pode auxiliar na melhoria de aspectos de Computação Gráfica e Visualização.

A seguir, uma breve descrição de cada aspecto que julgamos importante para que o ambiente seja melhorado com trabalhos futuros:

1. Computação Gráfica
   - qualidade das representações: outras técnicas de *rendering*;
   - construção/definição de outras representações: *Ribbons*, *Cartoon*, Superfície Energética;
   - interatividade: melhoramento das operações de rotação, escala e translação;
   - questões relativas a desempenho: otimização do *rendering* de múltiplos átomos na cena, *culling*.
2. Visualização
   - *dynamic queries*; novos filtros; filtros dinâmicos; filtros pré-definidos;
   - visualização multidimensional de átomos e aminoácidos e suas características através do uso de *Glyphs*;
   - agregar outras formas de visualização multidimensional, como *Glyphs*;
   - aplicar outras técnicas de Coordenadas Paralelas; Coordenadas Paralelas + Informações sobre as linhas horizontais; um *IRVE* de *Desktop* somente sobre as Coordenadas Paralelas, onde o usuário interage e aumenta/visualiza as informações existentes;
   - aspectos de *visual data mining*;

- aspectos de *feature extraction*;
- visualização comparativa: quantificação de diferenças (no caso de mapa de contatos);
- detecção da formação de estruturas secundárias em Mapas de Contatos e sua quantificação;
- visualização de contatos entre aminoácidos; representação visual dos contatos; configuração de valor mínimo para definição da existência de contatos.

3. Realidade Virtual

- gerenciador de painéis informativos, detecção e gerenciamento de oclusão;
- deslocamento automático; *way-finding*;
- navegação exploratória; navegar pelo ambiente; dicas de navegação;
- ampliação do uso da *SmallVR* no sistema; outras funcionalidades;
- contrução de ambiente imersivo de análise, visualização e interação.

4. IHC

- interação com usuário e com os dados; aproximação semântica; múltiplos contextos;
- interfaces; identificação de padrões por usuario (regiões mais utilizadas para análise);
- mostrar informações sem que o usuário as requisite, ou seja, um sistema que verifique o contexto e defina as melhores informações a serem exibidas; definição de regiões/níveis informativos (relaciona-se com aproximação semântica);
- reposição de contexto; navegação direcionada;
- avaliação de usuários.

5. Sistemas de Informação

- definição de uma *markup language* para *IRVE's*, definindo um formato padrão para enriquecimento de trajetórias de simulações de proteínas que seja hierárquica, genérica, temporal;
- integração de outras fontes de dados no ambiente; ampliação de informações prévias.

6. Processamento Paralelo e Distribuído

- cálculo de informações de cada passo em um *cluster*; paralelização de cálculos relevantes para o sistema;
- balanceamento de carga para as múltiplas simulações.

7. Outros

- uso da arquitetura para predição de estruturas;
- *web services*;
- conexão em servidores de proteínas, para salvamento e representação na cena de desenho.

Acabamos de sugerir possíveis trabalhos futuros para implementação no Ambiente *SimVIZ*. A Bioinformática Estrutural apresenta problemas desafiadores e este trabalho estabeleceu este fato. Por ser uma área multidisciplinar do conhecimento, mescla-se facilmente com muitas outras, como foi constatado acima, onde várias áreas da Ciência da Computação podem ser usadas para estender o ambiente e validar novos conceitos.

Referências

[1] A Font Library for OpenGL. Capturado em: http://plib.sourceforge.net/fnt/index.html, Novembro 2005.

[2] Amber 8 User's Manual. Capturado em: http://amber.scripps.edu/doc8/amber8.pdf, Outubro 2005.

[3] BC Online: 2C - Understanding Protein Conformation. Capturado em: http://employees.csbsju.edu/hjakubowski /classes/ch331/protstructure/olunderstandconfo.html, Outubro 2005.

[4] DNA DataBank of Japan. Capturado em: http://www.ddbj.nig.ac.jp/, Outubro 2005.

[5] Estrutura Terciária. Capturado em: http://www.emsl.pnl.gov /new/highlights/images/200410Fig4.jpg, Outubro 2005.

[6] ExPASy - Swiss-Prot and TrEMBL. Capturado em: http://ca.expasy.org/sprot/, Outubro 2005.

[7] NCBI HomePage - National Library of Medicine. Capturado em: http://www.ncbi.nlm.nih.gov/, Outubro 2005.

[8] Origin software. Capturado em: http://www.rockware.com/catalog/pages/origin.html, Outubro 2005.

[9] Part I: Introduction to Protein Structure. Capturado em: http://swissmodel.expasy.org/course/text/chapter1.htm, Outubro 2005.

[10] PDB Current Holdings. Capturado em: http://www.rcsb.org/pdb/holdings.html, Outubro 2005.

[11] PyMOL. Capturado em: http://www.pymol.org/, Outubro 2005.

[12] SmallVR - A Simple toolkit for VR Application Development. Capturado em: http://www.smallvr.org, Outubro 2005.

[13] The European Molecular Biology Laboratory. Capturado em: http://www.embl.org/, Outubro 2005.

[14] UniProtKB/Swiss-Prot Release 48.1 statistics. Capturado em: http://ca.expasy.org/sprot/relnotes/relstat.html, Outubro 2005.

[15] B. B. Bederson and J. D. Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. In *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 17–26, New York, NY, USA, 1994. ACM Press.

[16] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. GenBank. *Nucleic Acids Research*, 33 (Database Issue):D34–D36, 2005.

[17] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[18] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Research*, 31:365–370, 2003.

[19] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[20] D. A. Bowman, C. North, J. Chen, N. F. Polys, P. S. Pyla, and U. Yilmaz. Information-rich virtual environments: theory, tools, and research agenda. In *VRST '03: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 81–90, New York, NY, USA, 2003. ACM Press.

[21] C. Branden and J. Tooze. *Introduction to protein structure*. Garland, 1999.

[22] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[23] R. M. Czekster and O. N. de Souza. VIZ – A Graphical Open-Source Architecture for Use in Structural Bioinformatics. *Brazilian Symposium of Bioinformatics '2005. Lecture Notes in Bioinformatics*, 3594:226–229, 2005.

[24] M. C. F. de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):378–394, 2003.

[25] W. L. DeLano. The PyMOL Molecular Graphics System (2002). *DeLano Scientific, San Carlos, CA, USA*, 2002.

[26] C. M. Dobson. Protein Folding and Misfolding. *Nature*, 426:884–890, 2003.

[27] D. Frishman and P. Argos. Knowledge-based protein secondary structure assignment. *Proteins*, 23(4):566–579, 1995.

[28] Y. H. Fua, M. O. Ward, and A. Rundensteiner. Hierarchical Parallel Coordinates for Exploration of Large Datasets. *VISUALIZATION '99: Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)*, pages 43–51, 1999.

[29] N. Guex and M. C. Peitsch. SWISS-MODEL and the Swiss-PdbViewer: An environment for comparative protein modeling. *Electrophoresis*, 18:2714–2723, 1997.

[30] L. Holm and C. Sander. The FSSP database: fold classification based on structure-structure alignment of proteins. *Nucleic Acids Research*, 24(1):206–209, 1996.

[31] W. Humphrey, A. Dalke, and K. Schulten. VMD - Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.

[32] A. Inselberg and B. Dimsdale. Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry. *IEEE Visualization: Proceedings of the 1st conference on Visualization '90*, pages 361–378, 1990.

[33] A. D. Mackerell Jr, B. Brooks, C. L. Brooks, L. Nilsson, B. Roux, Y. Won, and M. Karplus. CHARMM: The Energy Function and Its Parameterization with an Overview of the Program. *The Encyclopedia of Computational Chemistry*, 1:271–277, 1998.

[34] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.

[35] L. Kalé, R. Skeel, M. Bhandarkar, R. Brunner, A. Gursoy, N. Krawetz, J. Phillips, A. Shinozaki, K. Varadarajan, and K. Schulten. NAMD2: Greater scalability for parallel molecular dynamics. *Journal of Computational Physics*, 151:283–312, 1999.

[36] C. Kanz, P. Aldebert, N. Althorpe, W. Baker, A. Baldwin, K. Bates, P. Browne, A. van den Broek, M. Castro, G. Cochrane, K. Duggan, R. Eberhardt, N. Faruque, J. Gamble, F. G. Diez, N. Harte, T. Kulikova, Q. Lin, V. Lombard, R. Lopez, R. Mancuso, M. McHale, F. Nardone, V. Silventoinen, S. Sobhany, P. Stoehr, M. A. Tuli, K. Tzouvara, R. Vaughan, D. Wu, W. Zhu, and R. Apweiler. The EMBL Nucleotide Sequence Database. *Nucleic Acids Research*, 33(suppl_1):D29–33, 2005.

[37] R. A. Laskowski, M. W. MacArthur, D. S. Moss, and J. M. Thornton. PROCHECK: a program to check the stereochemical quality of protein structures. *Journal of Applied Crystallography*, 26:283–291, 1993.

[38] A. R. Leach. *Molecular Modeling: Principles and Applications.* Person Education, 2001.

[39] N. M. Luscombe and D. Greenbaum e M. Gerstein. What is Bioinformatics? A Proposed Definition and Overview of the Field. *Methods of Information in Medicine*, 40:346–358, 2001.

[40] K. L. Ma. Visualization - A Quickly Emerging Field. *ACM SIGGRAPH Computer Graphics Quaterly*, 38(1):4–7, 2004.

[41] S. Miyazaki, H. Sugawara, K. Ikeo, T. Gojobori, and Y. Tateno. DDBJ in the stream of various biological data. *Nucleic Acids Research*, 32 (Database Issue):D31–D34, 2004.

[42] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman. AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications*, 91:1–41, 1995.

[43] M. S. Pinho. SmallVR: Uma Ferramenta Orientada a Objetos para o Desenvolvimento de Aplicações de Realidade Virtual. *Proceedings of Symposium on Virtual Reality '2002.*

[44] N. F. Polys and D. A. Bowman. Design and display of enhancing information in desktop information-rich virtual environments: challenges and techniques. *Virtual Reality*, 8(1):41–54, 2005.

[45] N. F. Polys, D. A. Bowman, and C. North. Information-Rich Virtual Environments: Challenges and Outlook. *NASA Virtual Iron Bird Workshop, 2004*, 2004.

[46] N. F. Polys, D. A. Bowman, C. North, R. Laubenbacher, and K. Duca. PathSim visualizer: an Information-Rich Virtual Environment framework for systems biology. In *Web3D '04: Proceedings of the ninth international conference on 3D Web technology*, pages 7–14, New York, NY, USA, 2004. ACM Press.

[47] W. Ribarsky, E. Ayers, J. Eble, and S. Mukherjea. Glyphmaker: Creating Customized Visualizations of Complex Data. *Computer*, 27(7):57–64, 1994.

[48] W. R. P. Scott, P. H. Hünenberger, I. G. Tironi, A. E. Mark, S. R. Billeter, J. Fennen, A. E. Torda, T. Huber, P. Krüger, and W. F. van Gunsteren. The GROMOS Biomolecular Simulation Program Package. *Journal of Physical Chemistry B*, 103:3596–3607, 1999.

[49] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages*, pages 336–343, Washington, DC, USA, 1996. IEEE Computer Society.

[50] D. Shortle. Prediction of Protein Structure. *Current Biology*, 10:R49–R51, 2000.

[51] A. Vailaya, P. Bluvas, R. Kincaid, A. Kuchinsky, M. Creech, and A. Adler. An architecture for biological information extraction and representation. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 103–110, New York, NY, USA, 2004. ACM Press.

[52] R. van Teylingen, W. Ribarsky, and C. van der Mast. Virtual Data Visualizer. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):65–74, 1997.

[53] M. Vendruscolo and E. Domany. Efficient dynamics in the space of contact maps. *Folding and Design*, 3(5):329–336, 1998.

[54] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, 2002.

Apêndice A

Manual de Usuário do Ambiente *SimVIZ*

Este capítulo apresenta os pré-requisitos, os detalhes de implementação, a interface gráfica e as principais funcionalidades do Ambiente *SimVIZ*.

A.1 Pré-requisitos

Os pré-requisitos do ambiente são os seguintes:

- Ambiente Windows;
- CPU com 1.0 Gigahertz;
- 64 Mb de memória RAM;
- a instalação ocupa 1.5 Mb (com alguns arquivos exemplo).

A.2 Detalhes de Implementação

A seguir alguns detalhes de implementação do Ambiente *SimVIZ*:

- Interface Gráfica construída em *FLTK - Fast Light Toolkit*;
- *OpenGL*;
- *IDE DevC++*;

- *SmallVR* [43];
- *STRIDE* [27];
- *FNT* - Pacote de fontes da *PLIB* [1].

A.3 Funcionalidades

Todas as funcionalidades são aplicadas a abertura de proteínas ou simulações. No caso de simulações, cada passo de simulação pode ser visto como uma proteína *stand-alone* e todas as funcionalidades são aplicadas a esta estrutura. Essas são as funcionalidades que o ambiente oferta aos usuários:

- Abertura de múltiplas trajetórias de simulação;
- Abertura de proteínas *stand-alone* no formato *PDB*;
- Interação com objetos na cena tridimensional: translação, rotação e redimensionamento;
- Visualização tridimensional de proteínas;
- Filtro por *backbone* do aminoácido;
- Visualizações disponíveis: *VDW*, *CPK*, *Lines* e *New_Ribbons* (esta última representação encontra-se em versão preliminar);
- Análise de proteínas:
  - Gráfico de Ramachandran - de uma estrutura de referência e de cada passo da simulação;
  - Mapa de Contatos - de uma estrutura de referência e de cada passo da simulação;
  - Gráficos bidimensionais de saídas de simulação produzidos pelo *AMBER*;
  - Gráfico de RMSD;
  - Visualização de Coordenadas Paralelas de saídas de simulação;
- Integração com o *STRIDE* [27] para descoberta de elementos estruturais secundários de proteínas;

A.4 Interface Gráfica de Usuário

Esta é a tela inicial do Ambiente *SimVIZ*. A Figura A.1 mostra a interface inicial do sistema. Acima, na figura, vemos a barra de menus, com as opções: *File*, *View* e *Help*. Ao centro, está a listagem de proteínas ou simulações abertas (no caso da figura, nenhuma foi aberta ainda). Abaixo estão os controles de simulação, com as seguintes opções: *Sleep* (onde define-se um intervalo entre cada passo de simulação, em milisegundos), *Play* (executa a trajetória de simulação, ou seja, cria uma animação da trajetória), *Stop* (para a execução da animação) e um controle *FLTK* para ir para um determinado tempo de simulação.

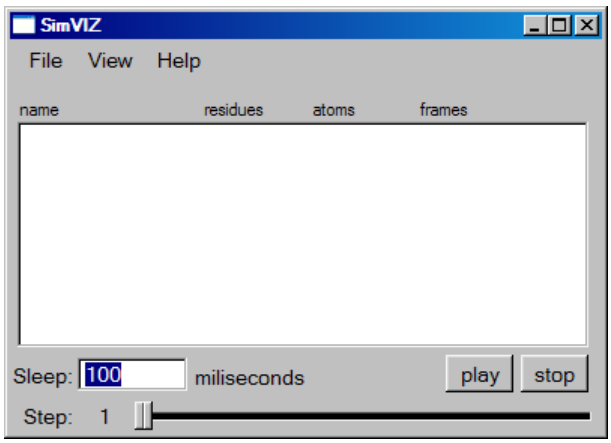

Figura A.1: Interface geral do Ambiente *SimVIZ*.

No menu *File*, descrito pela Figura A.2, as possibilidades de execução são: *Open* (abre uma proteína), *Open Simulation* (abre uma simulação), *Open RMSD* (abre um arquivo contendo valores de *RMSD*), *Open Ref. Structure* (abre a Estrutura de Referência da proteína), *Save Session* (salva as informações que foram associadas) e *Exit* (sai do ambiente).

O menu *View*, Figura A.3, apresenta as seguintes opções: *Representation* (abre um diálogo contendo as representações possíveis, filtros e formas de colorir para proteínas), *Information Manager* (abre a janela de associação de informações e exibição de gráficos e tabelas na cena tridimensional) e *Reset 3D View* (reinicia a cena tridimensional para uma posição padrão).

A Figura A.4 mostra a janela de visualização de proteínas e informações, criada a cada abertura de proteína ou simulação (uma para cada). Esta janela de visualização está configurada para criar e posicionar as luzes no ambiente gráfico e atualizar a proteína de acordo com as opções inseridas pelos usuários.

Para abrir simulações, o menu utilizado é o *File* > *Open Simulation*. A Figura A.5 mostra

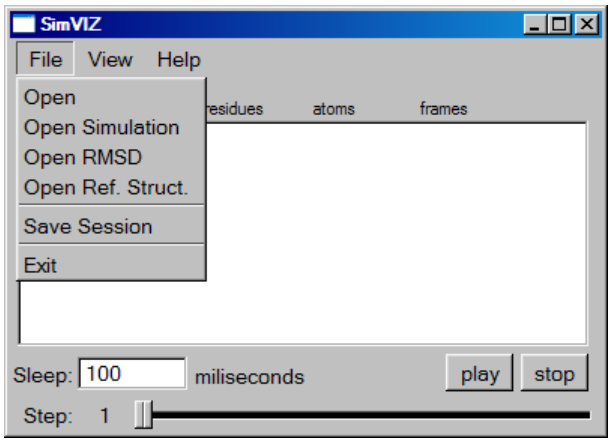

Figura A.2: Menu *File* do Ambiente *SimVIZ*.

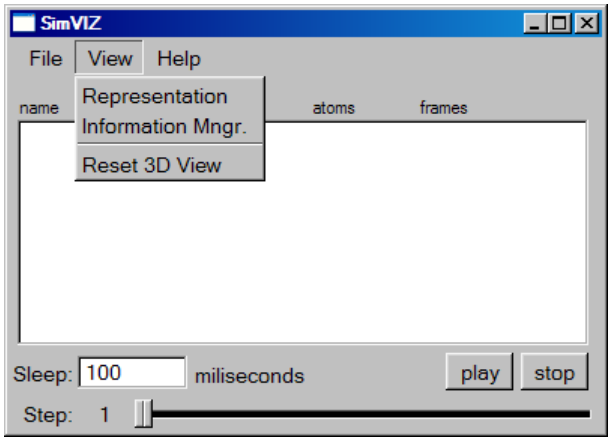

Figura A.3: Menu *View* do Ambiente *SimVIZ*.

a janela *FLTK* auxiliar que é criada quando o usuário realiza esta operação. A figura mostra 3 controles onde o usuário coloca o caminho (*Path*) da simulação, o padrão de nomes utilizado pela simulação (por exemplo, se a simulação possui os arquivos: 1zdb_1.pdb, 1zdb_1.pdb, 1zdb_1.pdb ... 1zdb_100.pdb o padrão de nomes seria '1zdb_%d.pdb'). Por fim, o usuário coloca o padrão de nomes para as saídas de simulação (para os arquivos existentes no formato *OUT*, seguindo a mesma idéia do padrão de nomes para a simulação, já que estes arquivos seguem a mesma lógica).

A Figura A.6 demonstra o menu *View* > *Representation*. Quando selecionado, mostra uma janela onde o usuário pode escolher um filtro, entre as opções *All* (tudo) e *Backbone* (somente o *backbone* da proteína). As demais opções são: escolher uma representação (entre as opções listadas acima), escolher uma forma de colorir, alterar atributos dos átomos e ligações atômicas

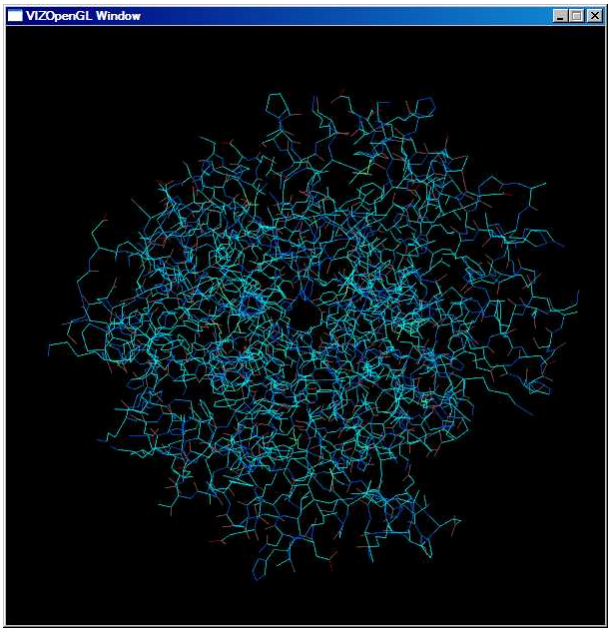

Figura A.4: Janela de visualização do Ambiente *SimVIZ*.

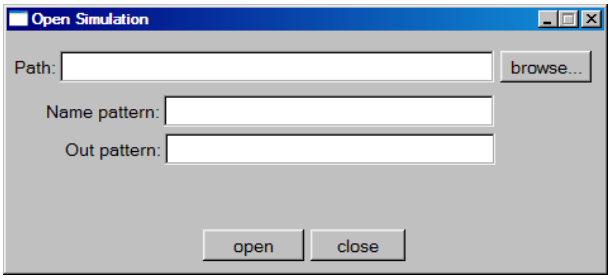

Figura A.5: Menu *File* > *Open Simulation* do Ambiente *SimVIZ*.

(alterando raios e diâmetros) e alterar a definição dos objetos gráficos (*Stacks* e *Slices*).

O menu *View* > *Information Manager*, mostrado pela Figura A.7, exemplifica a janela que coordena as informações na cena tridimensional, posicionando elementos gráficos, informações textuais, Coordenadas Paralelas, Gráfico de Ramachandran e Mapa de Contato.

Esta parte do ambiente permite a seleção das seguintes opções:

- General Data: mostra os dados globais relativos a simulação como um todo;
- Information: mostra os dados do passo de simulação e dos resíduos, caso existam;
- Atom Names: desenha os nomes do átomos, permitindo que se escolha o átomo a ter seu nome desenhado;

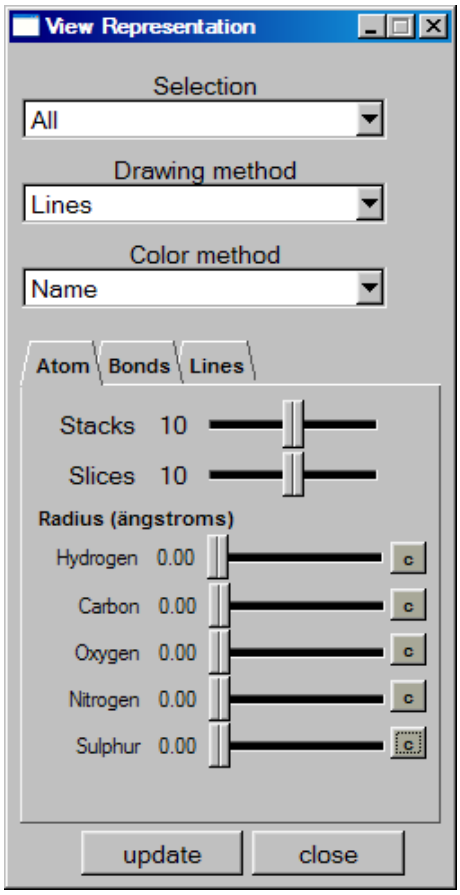

Figura A.6: Menu *View* > *Representation* do Ambiente *SimVIZ*.

- Atom Statistics: mostra as contagens sobre átomos de uma proteína;
- Ramachandran: representa o Gráfico de Ramachandran para o passo de simulação e para a Estrutura de Referência, caso esta tenho sido carregada;
- Sec. Structures: mostra rótulos sobre as estruturas secundárias utilizadas pelo *STRIDE*;
- Representation: desenha a proteína na cena de desenho tridimensionalmente;
- Residue List: coloca a lista de aminoácidos de uma proteína no painel informativo;
- RMSD Chart: Desenha o Gráfico de *RMSD* caso este tenha sido carregado pelo usuário;
- 2D Charts: mostra os gráficos bidimensionais das saídas de simulação;
- 2D Charts + data: mostra os gráficos bidimensionais das saídas de simulação, incluindo outras informações, como valores máximos, mínimos, desvio padrão e média;

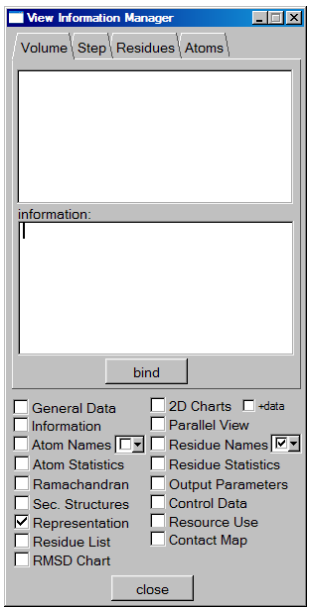

Figura A.7: Menu *View* > *Information Manager* do Ambiente *SimVIZ*.

- Parallel View: desenha a representação multidimensional conhecida por Coordenadas Paralelas na cena de desenho, para os passos de simulação, desenhando as informações de saída de simulações;
- Residue Names: mostra os nomes dos resíduos, com filtro de acordo com o aminoácido desejado;
- Residue Statistics: representa as contagens sobre os aminoácidos de cada tipo;
- Output Parameters: desenha os parâmetros de saídas de simulação;
- Control Data: desenha os parâmetros de controle do *AMBER*;
- Resource Use: mostra informações do *AMBER*;
- Contact Map: desenha o Mapa de Contatos da proteína no passo de simulação e o Mapa de Contatos da Estrutura de Referência caso ela tenha sido carregada pelo usuário.

Esta janela ainda permite que sejam associadas informações globais sobre a simulação, informações sobre os passos de simulação e sobre os aminoácidos individuais de cada tempo de simulação, que serão salvos em um arquivo especial (no menu *File* > *Save Session*).

A Figura A.7 mostra, na região superior da janela, 4 'abas' (*Volume* - informações globais, *Step* - informações relativas ao passo de simulação, *Residues* - relativas aos aminoácidos e *Atoms*

- relativa aos átomos, este último não é utilizado) e 1 botão chamado *Bind*. Nas abas são escolhidas as formas de associação de informação e o botão salva esses dados em estruturas de dados internas do ambiente, para recuperação futura e apresentação na cena de desenho.

Caso exista uma informação previamente associada, o ambiente recupera e mostra na janela, assim o usuário pode editar a informação, alterando os valores anteriores.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Administração
Baixar livros de Agronomia
Baixar livros de Arquitetura
Baixar livros de Artes
Baixar livros de Astronomia
Baixar livros de Biologia Geral
Baixar livros de Ciência da Computação
Baixar livros de Ciência da Informação
Baixar livros de Ciência Política
Baixar livros de Ciências da Saúde
Baixar livros de Comunicação
Baixar livros do Conselho Nacional de Educação - CNE
Baixar livros de Defesa civil
Baixar livros de Direito
Baixar livros de Direitos humanos
Baixar livros de Economia
Baixar livros de Economia Doméstica
Baixar livros de Educação
Baixar livros de Educação - Trânsito
Baixar livros de Educação Física
Baixar livros de Engenharia Aeroespacial
Baixar livros de Farmácia
Baixar livros de Filosofia
Baixar livros de Física
Baixar livros de Geociências
Baixar livros de Geografia
Baixar livros de História
Baixar livros de Línguas

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo