

MIG-SSM: MÓDULO DE INTERFACE GRÁFICA DE UM SISTEMA DE SIMULAÇÃO MOLECULAR – APLICAÇÕES COM O PROGRAMA LMPROT

Paulo R. Almeida¹, Shindy T. Miura², Sandra H. Oliveira³ e Roosevelt A. da Silva⁴

¹⁻⁴Universidade do Vale do Paraíba - Faculdade de Ciência da Computação
Av. Shishima Hifumi, 2911 CEP 12244-000 São José dos Campos - SP – Brasil

[1paulorobertoalmeida@ig.com.br](mailto:paulorobertoalmeida@ig.com.br), [2stmiura@ig.com.br](mailto:stmiura@ig.com.br), [3sandra_oliver@ig.com.br](mailto:sandra_oliver@ig.com.br), [4roossilva@univap.br](mailto:roossilva@univap.br)

Resumo: Neste trabalho foi desenvolvido um módulo de Interface Gráfica (MIG) para programas de simulação molecular que, em particular, foi aplicado ao programa LMProt. A interface facilita a manipulação e a geração dos parâmetros de entrada do programa e gera um arquivo com todos os parâmetros necessários para a simulação. Além disso, a interface executa o programa desejado, na plataforma LINUX, e lê, automaticamente, de tempos em tempos, todos os arquivos de saída e os grafa de acordo com o critério do usuário. A interface foi desenvolvida na plataforma Java, utilizando a ferramenta NetBeans 5.0 e será incorporada como um módulo do SSM (Sistema de Simulação Molecular).

Palavras-chave: Interface Gráfica, Java, Programas de Simulação Molecular, Enovelamento de Proteínas.

Área do Conhecimento: Ciências Exatas e Biológicas.

Introdução

Proteínas são macromoléculas que estão envolvidas em praticamente todos os processos vitais dos seres vivos. Ao lado dos ácidos nucleicos, as proteínas configuram o principal alvo de estudo da biologia molecular. Elas são alvos importantes principalmente para o desenvolvimento de drogas e no entendimento de diversas doenças, pois são capazes de inibir ou ativar outras moléculas. Na atual era proteômica, decifrar as estruturas 3D, e o mecanismo de interação e de enovelamento das proteínas, trata-se de um dos temas centrais de interesse biotecnológico principalmente para a indústria farmacêutica [1].

As proteínas em geral são moléculas de estruturas 3D que são sintetizadas e formadas pela molécula de ribossomo, que no interior da célula, assumem uma estrutura 3D específica de baixa energia, cuja especificidade relaciona-se diretamente com sua função no organismo. Elas são classificadas em diferentes tipos de estruturas: a estrutura primária que representa apenas a seqüência de aminoácidos contidos na proteína, a estrutura secundária que determina a presença de conformações regulares locais, a estrutura terciária que representa a distribuição no espaço tridimensional dos aminoácidos, contidos na proteína, e a estrutura quaternária que representa a distribuição espacial de proteínas com mais de uma cadeia polipeptídica [2]. Na natureza as proteínas se apresentam em suas estruturas terciárias e quaternárias, as quais, como já dito, só desempenham suas funcionalidades quando assumem estruturas desta natureza.

O processo para determinar a seqüência de aminoácidos em uma proteína é relativamente simples, mas nem sempre é possível determinar a estrutura tridimensional de uma proteína, pois os métodos existentes para esse fim (Cristalografia e Ressonância Magnética Nuclear e outros) dependem de uma série de condições que podem não ser atingidas facilmente [3].

Dessa forma, a obtenção de um modelo tridimensional de uma proteína a partir da seqüência primária de resíduos, como também o entendimento das interações entre proteína-proteína e proteína-ligantes, por meio de técnicas computacionais, pode tornar-se uma ferramenta indispensável para a elucidação rápida dos diversos fenômenos biológicos que ainda não foram completamente desvendados.

O presente trabalho trata-se de uma ferramenta computacional desenvolvida com a finalidade de facilitar os procedimentos usuais que em geral estão presentes nos programas de simulação molecular, como a designação das especificidades das moléculas (tipo da proteína ou ligantes), campo de força, parâmetros envolvidos na simulação tipo Monte Carlo e os nomes e diretórios envolvendo os arquivos de entrada e saída do programa de simulação.

A Interface Gráfica é um mecanismo que facilita a interação entre o usuário e o programa. Através de um dispositivo de entrada (mouse ou teclado) o usuário é capaz de selecionar as opções desejadas de forma a obter algum resultado prático.

Uma ferramenta poderosa para a construção de Interfaces Gráficas é o NetBeans, que é uma ferramenta visual e interativa, permitindo a fácil criação de programas Java,

C/C++, XML, HTML e muitas outras linguagens. Seu editor visual de interface gráfica Swing é um dos mais usados neste tipo de aplicação pelas facilidades no desenvolvimento em J2EE. A tecnologia de componentes permitiu o desenvolvimento de complexos sistemas de informação através da combinação e extensão de blocos reutilizáveis de software. Nessa linha, os JavaBeans emergiram como um importante padrão de componentes Java para aplicações cliente, com as vantagens de portabilidade e desenvolvimento em ferramentas visuais.

Materiais e Métodos

Neste trabalho, para o desenvolvimento da Interface gráfica requerida, utilizou-se a ferramenta NetBeans 5.0 por facilitar a codificação do aplicativo na linguagem Java [4], visto que o acesso a bibliotecas de construção de interfaces tipo janelas e Pacote AWT, que contém as classes e interfaces exigidas para criar e manipular interfaces gráficas com o usuário em Java, é obtido diretamente.

O Módulo de Interface Gráfica (**MIG**) construída torna o procedimento de entrada dos parâmetros iniciais da simulação amigável e permite que o usuário tenha também acesso às informações dos arquivos de saída das simulações por meio de gráficos que o próprio módulo gera automaticamente, atualizando-os conforme o andamento das simulações.

O desenvolvimento do módulo *MIG-SSM* foi dividido em 4 partes:

- a interface gráfica;
- a geração de relatórios;
- a chamada e execução do programa de simulação molecular;
- a aquisição dos dados de saída do programa para a geração dos gráficos desejados pelo usuário.

As caixas de entrada permitem que o usuário entre com os parâmetros desejados para a simulação a fim de serem lidos e convertidos no formato do arquivo de entrada requerido pelo programa de simulação molecular, que neste trabalho em especial foram preparadas para a geração do arquivo de entrada do programa LMProt [5].

A execução do programa executável desejado é realizada através de um menu que chama o programa abrindo uma nova shell e o usuário tem a opção de chamar o executável do LMProt, ou qualquer outro programa pelo usuário definido. No mesmo menu temos a opção dos gráficos.

As gerações dos gráficos ocorrem através dos arquivos de saída do LMProt. O usuário pode

definir o arquivo e quais colunas serão grafadas. Não existe um limite, o usuário pode gerar quantos gráficos quiser em tempo real. As funções gráficas foram construídas utilizando a classe Graphics, pois a mesma, nos permite desenhar linhas, formas geométricas e texto. Existe dois pacotes importantes nesta classe: Pacote Swing e Pacote AWT. Porém, foram utilizados mais os componentes Swing, por serem praticamente todos escritos em Java, o que nos forneceu maior nível de portabilidade e de flexibilidade.[6]

Resultados

Todos os pré-requisitos do MIG foram completamente atendidos de modo que o resultado final foi um produto amigável, isto é, de fácil uso, e com robustez (código fonte) o suficiente para ser empregado em outros programas de simulação que não seja especificamente o LMProt. O MIG foi testado sob a plataforma LINUX (Fedora) em micros do Laboratório da Faculdade de Ciência da Computação / UNIVAP. A seguir são mostrados alguns dos itens concluídos até o momento pelo MIG:

1. Interface (Fig.1)

- Criação dos menus;
- Desenvolvimento dos itens necessários para entrada de dados (Fig.2);
 - a) Iseeds;
 - b) Models;
 - c) LMProt;
 - d) Pivot;
 - e) MonteCarlo;
- Tratamento de dados:
 - a) Se usuário definir algum dado como desnecessário (not) a caixa de texto para informações adicionais fica desabilitada;
- Definições de diretórios para:
 - a) Arquivo Executável;
 - b) Arquivos dos Gráficos;
 - c) Arquivo de Relatório;
- Recuperação dos últimos dados como *default* para as definições de diretórios;
- Geração de relatório;
- Visualização de relatório;

2. Gráficos (Fig.3)

- Definição de arquivo;
- Opção de gerar um gráfico para uma ou várias colunas dos arquivos de saída, sem necessitar fecharem demais gráficos;

3. Executável

Opção de chamada para o executável do LMProt através da Interface.

molecular. O gráfico é atualizado em tempo real e re-escalado dinamicamente.

4. Realização de Testes e Simulações;

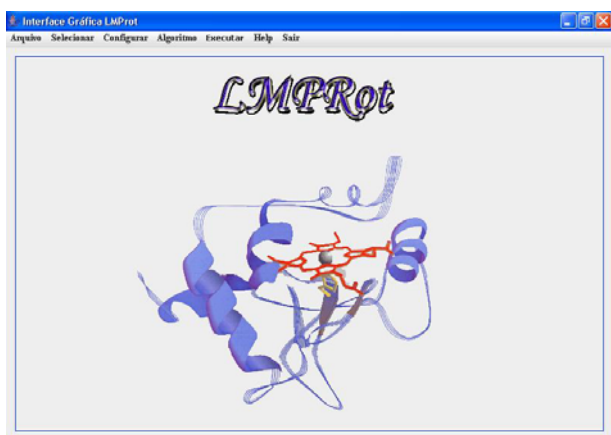


Fig.1 - Tela Principal do MIG. A interface gráfica contém 7 menus: Arquivo (Gerar arquivo e Visualizar Arquivo), Selecionar (Configura Diretório), Configurar (Iseeds e Model), Algoritmo (LMPRot, Pivot e Monte Carlo), Executar (LMPRot e Gráfico), Help e Sair (Finalizar.).

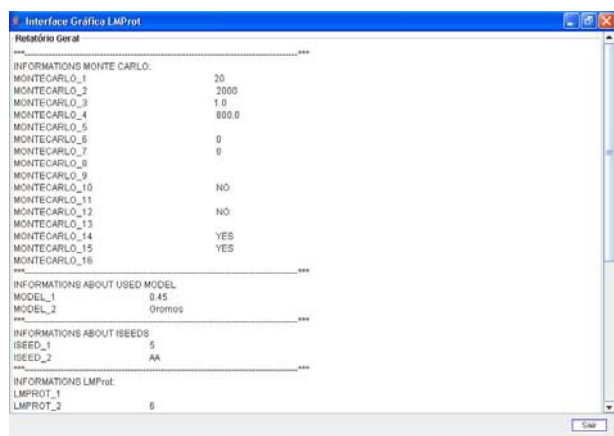


Fig.2 – Relatório: dados de configuração gerados pelo MIG num arquivo de entrada para o programa de simulação molecular LMProt.

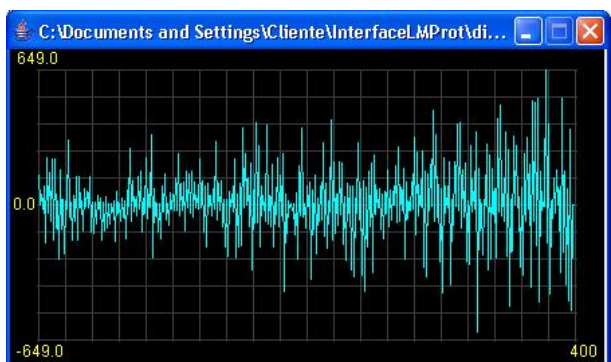


Fig.3 – Gráfico gerado a partir dos dados obtidos durante a execução do programa de simulação

Conclusão

Todos os pré-requisitos solicitados para a geração correta do arquivo de entrada do programa LMProt foram satisfeitos. A chamada ao programa executável a partir do arquivo de entrada, construído pelo MIG, procedeu-se com sucesso sob a plataforma LINUX e todos os resultados gerados pelo programa, ou seja, os arquivos de saída foram corretamente lidos e grafados adequadamente.

Pretensões Futuras

Embora a premissa para a construção do MIG fosse obter um produto genérico, capaz de adequar-se a qualquer outro programa de simulação molecular, a concepção dos itens de cada menu, neste trabalho em particular, foi dirigida para aplicações com o programa LMProt. Considerando que os testes realizados com o LMProt foram bem sucedidos, o próximo passo será tornar o atual MIG capaz de, interativamente, para os administradores do software, conceber os menus de modo que os critérios requeridos pelos arquivos de entrada sejam definidos diretamente no próprio MIG.

Adicionalmente, é pretendido que o MIG seja incorporado pelo SSM (sistema de simulação molecular) que atualmente está em desenvolvimento pelo Grupo de Computação Biomédica da UniVap, o qual além do MIG envolverá o módulo de sistema distribuído (MSD) e o Módulo de Computação Paralela (MCP).

Referências

- [1] Fernández-Recio J, Totrov M and Abagyan R. *Identification of Protein-Protein Interaction Sites from Docking Energy Landscapes*. J. Mol. Biol. **335**: 843–865 (2004).
- [2] Mathews CK, van Holde KE and Ahern KG. *Biochemistry*. Benjamin Cummings (2000).
- [3] Pain RH. *Mechanisms of protein folding*. Oxford University Press, New York (1994).
- [4] Java 2 – Fundamentos, Swing e JDBC. Ivan Mecnas, Editora Alta Books
- [5] daSilva RA, Degrève L and Caliri A. *LMProt: An Efficient Algorithm for Monte Carlo Sampling of*

Protein Conformational Space. *Biophysical Journal* **87**:1567-1577 (2004).

- [6] Programando em Java 2 – Interfaces gráficas e Aplicações práticas.
Osmar J. Silva, Editora Erica.