PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

SELEÇÃO EFICIENTE DE CONFORMAÇÕES DE RECEPTOR FLEXÍVEL EM SIMULAÇÕES DE DOCAGEM MOLECULAR

KARINA DOS SANTOS MACHADO

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Osmar Norberto de Souza

Porto Alegre
2011

Dados Internacionais de Catalogação na Publicação (CIP)

M149s Machado, Karina dos Santos
Seleção eficiente de conformações de receptor flexível em simulações de docagem molecular / Karina dos Santos Machado. – Porto Alegre, 2011.
180 f.

Tese (Doutorado) – Fac. de Informática, PUCRS.
Orientador: Prof. Dr. Osmar Norberto de Souza.

1. Informática. 2. Mineração de Dados (Informática). 3. Biologia Molecular. 4. Banco de Dados. I. Souza, Osmar Norberto. II. Título.

CDD 005.74

Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE TESE DE DOUTORADO

Tese intitulada "Seleção Eficiente de Conformações de Receptor Flexível em Simulações de Docagem Molecular", apresentada por Karina dos Santos Machado, como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação, Bioinformática e Modelagem Computacional, aprovada em 31/03/2011 pela Comissão Examinadora:

Prof. Dr. Osmar Norberto de Souza – PPGCC/PUCRS
Orientador

Prof. Dr. Laurent Emmanuel Dardenne - LNCC - RJ

Prof. Dr. Hermes Luís Neubauer de Amorim - ULBRA

Prof. Dr. Adriano Velasque Werhli - FURG

Prof. Dr. Duncan Dubugras Alcoba Ruiz – PPGCC/PUCRS

Homologada em 14./06./11...., conforme Ata No. 10.... pela Comissão Coordenadora.

Prof. Dr. Fernando Luís Dotti
Coordenador.

PUCRS
Campus Central
Av. Ipiranga, 6681 – P. 32 – sala 507 – CEP: 90619-900
Fone: (51) 3320-3611 – Fax (51) 3320-3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/facin/pos

DEDICATÓRIA

Dedico a Tese para minha família. Em especial para meus avós Elzira e Clodomiro.

Se você quiser alguém em quem confiar, confie em si mesmo. Quem acredita sempre alcança.

Renato Russo

AGRADECIMENTOS

Agradecer é o mínimo que eu posso fazer a muitas pessoas que de uma forma ou outra contribuíram para que eu chegasse ao fim dessa longa jornada. Primeiramente agradeço ao Programa de Pós-Graduação em Ciência da Computação da PUCRS e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) que me concederam a bolsa integral que permitiu que eu realizasse esse Doutorado.

Agradeço ao meu orientador Prof. Osmar Norberto de Souza pela oportunidade, pelas lições aprendidas e pelo tempo dedicado a este trabalho ao longo dos 4 anos. Com o Prof. Osmar aprendi como ser pesquisadora e entendi a dedicação que esta carreira exige.

Preciso agradecer, e muito, ao Prof. Duncan Ruiz, para mim um exemplo de pessoa e profissional, o qual eu certamente vou me espelhar a partir de agora na minha carreira acadêmica. Obrigada pelas discussões, pelas conversas e pelas palavras de incentivo.

Um parágrafo de agradecimentos precisa ser dedicado aos amigos que fiz neste período. Agradeço a minha grande amiga e principal parceira nesta pesquisa, Ana Trindade Winck. Com a Ana aprendi muito e compartilhei os melhores e os piores momentos. Também agradeço muito a Elisângela Cohen, convivência sempre agradável e divertida, cujas palavras de incentivo certamente fizeram muita diferença nos momentos difíceis, além da ajuda com as correções e com as dúvidas sobre Biologia sempre que solicitado. Agradeço também aos demais colegas e ex-colegas do LABIO, Renata de Paris, Ivani Pauli, Anderson Amaro, André Luis e Christian Quevedo. Em especial à sempre alegre Danieli Forgiarini com suas histórias divertidas, à Carla Aguiar com seu carinho e à Furia Gargano, com seus conselhos. Obrigada colegas, por tudo.

Agradeço muito ao coordenador do Centro de Ciências Computacionais da Universidade Federal do Rio Grande (FURG), Prof. Nelson Duarte Filho. Sua compreensão permitiu que eu concluisse o Doutorado. Também agradeço muito aos demais amigos e colegas da FURG, em especial a Danúbia Espíndola, Rafael Penna, Odorico Mendizabal, Cristina Meinhart, André Vargas, Sílvia Botelho e Diana Adamatti. O apoio e incentivo de todos foi muito importante. Aos meus amigos de sempre, Greyce Schroeder, Sidnei Franco, Diego Gomes, e demais ex-colegas da EC7, obrigada pela torcida.

O último e mais especial agradecimento é para minha família. Aos meus pais, Paulo e Clarisse, meu exemplo e maior motivação para concluir esse trabalho. Às minhas queridas e amadas irmãs Ana e Raquel e ao meu irmãozinho Renan, meus melhores amigos, obrigada por tudo. Mesmo a distância, sempre acreditaram que eu chegaria ao fim. Muito obrigada também a toda turma de netos da Vó Zira.

SELEÇÃO EFICIENTE DE CONFORMAÇÕES DE RECEPTOR FLEXÍVEL EM SIMULAÇÕES DE DOCAGEM MOLECULAR

RESUMO

O desenvolvimento de fármacos é um dos grandes desafios da ciência atual por se tratar de um processo onde os custos e o tempo envolvido são elevados. Um dos problemas mais interessantes nessa área é a predição da conformação e da energia envolvida na interação entre ligantes e suas proteínas-alvo ou receptores. É nos experimentos de docagem molecular que essa interação é avaliada. É muito comum que durante a docagem molecular se façam simplificações onde o receptor é tratado como rígido. Porém, proteínas são inerentemente sistemas flexíveis e essa flexibilidade é essencial para a sua função. A inclusão da flexibilidade do receptor em experimentos de docagem molecular não é uma tarefa trivial, pois, para permitir mobilidade a certos átomos do receptor, há um aumento exponencial do número de graus de liberdade a serem considerados. Há atualmente diversas alternativas para contornar esse problema, entre elas, a que se optou neste trabalho: considerar a flexibilidade explícita do receptor por meio da execução de uma série de simulações de docagem molecular, utilizando em cada um deles uma conformação diferente da trajetória dinâmica do receptor, gerada por uma simulação por dinâmica molecular (DM). Um dos maiores problemas desse método é o tempo necessário para executá-lo. Sendo assim, o objetivo desse trabalho é contribuir para a seleção de conformações do receptor de forma a acelerar a execução de experimentos de docagem molecular com o receptor completamente flexível. Além do mais, o trabalho apresenta novas metodologias para a análise da interação receptor-ligante em simulações de docagem deste tipo. Para alcançar esses objetivos, é aplicado um processo de descoberta de conhecimento. A primeira etapa consistiu no desenvolvimento de um banco de dados para armazenar informações detalhadas sobre o receptor e suas conformações, ligantes e experimentos de docagem molecular, chamado FReDD. Com os dados organizados no FReDD, foi possível a aplicação de diferentes técnicas de mineração de dados. O primeiro conjunto de experimentos foi realizado utilizando o algoritmo de classificação J48. O segundo conjunto de experimentos foi executado com o algoritmo de regressão M5P, onde apesar de resultados interessantes, a utilização direta para seleção de conformações em futuros experimentos de docagem molecular não se mostrou promissora. Finalmente, foram executados os experimentos de agrupamento com 10 diferentes algoritmos, com entradas variadas. Para os algoritmos de agrupamento foram desenvolvidas diferentes funções de similaridade onde os resultados finais utilizados em conjunto com o padrão de dados P-MIA permitiu a redução efetiva da quantidade de experimentos de docagem.

Palavras-chave: Mineração de Dados, Docagem Molecular, Receptor Flexível, Dinâmica Molecular.

EFFICIENT SELECTION OF FLEXIBLE RECEPTOR SNAPSHOTS APPLIED IN MOLECULAR DOCKING SIMULATIONS

ABSTRACT

Drug Development is one of the biggest challenges of current science since it deals with a process involving time and high costs. One of the most interesting problems in this area is the conformation and energy prediction between ligand and target proteins (or receptors) interaction, where such interaction is evaluated through molecular docking. It is very common to make simplifications such as to treat the receptor structure as rigid during a molecular docking. However, proteins are inherently flexible, and its flexibility is essential for its function. The inclusion of receptor flexibility in docking experiments is not a trivial task, since the allowance of mobility to some receptor atoms implies in an exponential increase in the numbers of degrees of freedom to be considered. Nowadays there are a variety of alternatives to treat this problem, as such the one chosen for this work: to consider the receptors explicit flexibility through a series of molecular docking simulations, using in each one, one different conformation (or snapshot) from a dynamic trajectory, generated by a molecular dynamic simulation (MD). This method execution, however, has the disadvantage of being very time-consuming. In doing so, the aim of this work is to contribute to the selection of receptors conformations in order to execute docking experiments faster, still taking into account the fully receptors flexibility. Besides, this work introduces new methodologies to analyze receptor-ligand interaction in this kind of docking simulations. To achieve this, it is applied a Knowledge Discovery in Databases (KDD) process. The first step required the development of a database, called FReDD. Such a database store detailed information about the receptors and its conformations, ligands and molecular docking results. From the data stored on FReDD, it was possible to apply different data mining techniques. The first set of experiments was performed with the J48 classification algorithm. The second one was executed using M5P regression algorithm, where despite the interesting results, the application of the induced models directly on snapshot selection seems to be not promising. Finally, clustering experiments were executed with 10 different algorithms with a variety of inputs. For these clustering algorithms, we developed different similarity functions where the final results, combined with the P-MIA data pattern, allowed the effective reduction in the amount of docking experiments to be performed.

Keywords: Data Mining, Molecular Docking, Flexible Receptor, Molecular Dynamics.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

3D	Tridimensional
AMBER	*Assisted Model Building with Energy Refinement*
ARFF	*Attribute-Relation File Format*
AutoDock3.0.5	*Automated Docking*
BD	Banco de Dados
DBI	*Davies-Bouldin Index*
DM	Dinâmica Molecular
DP	Desvio Padrão
EM	*Expectation-Maximization*
ETH	Etionamida
FDA	US *Food and Drug Administration*
FEB	*Free Energy of Binding* ou Energia Livre de Ligação
FFR	*Fully-Flexible Receptor* ou Receptor completamente flexível
FReDD	*Flexible Receptor Docking Database*
FReDoWS	*Flexible-Receptor Docking Workflow System*
GPIN	Grupo de Inteligência em Processo de Negócios
InhA	Enzima *2-trans-enoil ACP(CoA) Redutase* de *Mycobacterium tuberculosis*
IPF	Isoniazida pentacianoferrato
KDD	*Knowledge Discovery in Databases*
LABIO	Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas
LGA	*Lamarckian genetic algorithm*
LM	*Linear Mode*
MAE	*Mean Absolute Error*
NADH	Nicotinamida Adenina Dinucleotídeo, forma reduzida
PDB	*Protein Data Bank*
PIF	*Pentacyano(isoniazid)ferrate* II
pSF	*pseudo F-statistic*
RCS	*Relaxed Complex Scheme*
RDD	*Rational Drug Design*
RMSD	*Root Mean Square Deviation*
RMSE	*Root Mean Squared Error*

SGBD	Sistema Gerenciador de Banco de Dados
SOM	*Self-Organizing Maps*
SQL	*Structured Query Language*
TCL	Triclosano
VSL	*Virtual Screening* ou Triagem Virtual
WEKA	*Waikato Environment for Knowlodge Analysis*

SUMÁRIO

1. INTRODUÇÃO

1.1 Caracterização do Problema

Um dos grandes desafios da ciência hoje é sem dúvida o desenvolvimento de novos fármacos [ALO06]. Trata-se de um processo complexo e interdisciplinar, dirigido por esforços combinados da indústria farmacêutica, companhias de biotecnologia, autoridades reguladoras, pesquisadores acadêmicos e outros setores privados e públicos. Além disso, os custos envolvidos são muito altos: em média um bilhão de dólares até a aprovação de um novo fármaco [ADM10], assim como o tempo despendido é em torno de 10 a 15 anos [CAS07].

Existe uma grande variedade de abordagens computacionais que podem ser aplicadas aos diferentes estágios do processo de desenvolvimento de novos fármacos. Nos primeiros estágios, o foco está em reduzir o número de ligantes cuja interação com o receptor provavelmente será favorável, enquanto que, nos passos finais, os esforços estão direcionados em diminuir os custos experimentais e reduzir o tempo de execução dos experimentos.

Enquanto alguns pesquisadores de novos fármacos focam em soluções alternativas para otimizar o processo, outros trabalham na melhoria dos protocolos já existentes. Essas melhorias podem ser direcionadas para incorporar a flexibilidade da proteína em simulações de docagem molecular, explorar extensivamente as conformações do ligante dentro do sítio ativo, refinar e estabilizar o complexo receptor-ligante final, estimar as energias livres de ligação, entre outras [ALO06]. Segundo Broughton [BRO00], um dos desafios mais interessantes na área de desenvolvimento de novos fármacos está justamente relacionado com a predição da geometria e da energia envolvida na interação entre ligantes em suas proteínas-alvo.

É nas simulações de docagem molecular que a interação entre um receptor[1] e um ligante[2] é analisada. É muito comum que durante a docagem molecular se façam simplificações drásticas, onde tipicamente, o receptor é tratado como rígido [HUA06].

Porém, quando somente uma conformação do mesmo é considerada, de acordo com Totrov & Abagyan [TOT08], os algoritmos atuais de docagem molecular predizem erroneamente o local de ligação para 50 a 70% dos ligantes analisados. Ademais, proteínas são sistemas inerentemente flexíveis [COZ08] sendo essa flexibilidade essencial para determinar sua função. Além disso, de todas as possíveis conformações que uma proteína pode apresentar em determinado intervalo de tempo, não é possível conhecer *a priori* qual ou quais dessas conformações será adotada em resposta a ligação a determinado ligante ou como desenhar um ligante para uma conformação ainda desconhecida [COZ08].

A inclusão da flexibilidade do receptor em simulações de docagem molecular não é uma tarefa trivial. Para permitir mobilidade a certos átomos do receptor há um aumento exponencial no

[1] No presente trabalho macromolécula, proteína e receptor são termos tratados com o mesmo significado.
[2] Ligantes, inibidores e pequenas moléculas são termos tratados como sinônimos no texto.

número de graus de liberdade a serem considerados [WON08] pelos algoritmos e, conseqüentemente, uma exploração por força bruta se torna impossível de ser executada em um tempo de execução razoável [FER04].

Há atualmente diversas alternativas para contornar esse problema. Por exemplo: pode-se utilizar bibliotecas de rotâmeros para as cadeias laterais [LEA94], diferentes estruturas cristalográficas do receptor [MOR98, CLA01], uma estrutura média baseada em um conjunto de estruturas [KNE97] ou ainda um conjunto de conformações do receptor resultantes de uma simulação por dinâmica molecular (DM) [LIN02, MAC07]. Alguns trabalhos estão relacionados somente com a consideração do movimento de cadeias laterais do sítio de ligação no receptor, pelo algoritmo de docagem [CAR00] ou por meio da indicação do usuário, de quais cadeias laterais, ou partes da cadeia principal, ele deseja considerar como flexíveis durante as simulações de docagem [ZHA07], entre muitas outras alternativas revisadas por Alonso *et al.* [ALO06], Cozzini *et al.* [COZ08], B-Rao *et al.* [BRA09] e Wong [WON08].

Entretanto, um dos métodos mais promissores para o tratamento da flexibilidade do receptor é por meio da geração de subconjuntos de conformações do mesmo pela simulação por DM [van90]. Esse é considerado hoje o método mais acessível de se produzir muitas conformações da proteína com um custo razoável [COZ08]. Com essa metodologia uma série de conformações, denominada trajetória dinâmica do receptor, é gerada e utilizada em estudos de docagem molecular [LIN02, LIN03]. Dessa maneira, pode ser executado um conjunto de simulações de docagem molecular, utilizando-se em cada uma, uma conformação diferente da trajetória dinâmica. Neste trabalho,foi adotada a nomenclatura de modelo FFR (*Fully-Flexible Receptor*) para simulações de docagem com a consideração explícita da flexibilidade do receptor.

Um dos maiores problemas desse método é o tempo necessário para executá-lo [WON08, COZ08]. Um exemplo do tempo de execução para a consideração do modelo FFR de receptores em docagem molecular seria: dado um banco de dados de pequenas moléculas (ligantes) como o ZINC [IRW05] que atualmente tem mais de 20 milhões de compostos disponíveis. A análise da interação de todos esses possíveis candidatos, mesmo que *in-silico*, com uma determinada proteína-alvo (para cada proteína considera-se que seu modelo FFR tem 3.000 diferentes conformações) se torna inviável de ser executada, pois, estima-se que seriam necessários em torno de 650.000.000 hs (mais de 74 mil anos) até o término da execução desse experimento (onde cada experimento proteína-ligante é executado em aproximadamente 1 minuto em uma máquina Core2Quad, 2,4 GHz, 8 GB de memória RAM). Por essa razão, é essencial que se pesquisem maneiras menos custosas de incorporar a flexibilidade dos receptores nas simulações de docagem molecular.

A proposta desse trabalho é contribuir para um melhor entendimento da interação receptor-ligante em simulações de docagem molecular com o modelo FFR utilizando um processo de descoberta de conhecimento em banco de dados (KDD, do inglês *Knowledge Discovery in Databases*). A partir do entendimento detalhado da forma de interação do modelo FFR-ligante é possível realizar a seleção de conformações do receptor de forma a acelerar a execução de experimentos de docagem desse tipo.

Para alcançar os objetivos propostos, como primeira etapa, tornou-se necessário o desenvolvimento do FReDD (*Flexible Receptor Docking Database*), um banco de dados para armazenar informações sobre o receptor e suas conformações, ligantes e simulações de docagem molecular. Assim, os dados foram organizados de tal maneira que se tornou possível utilizá-los em experimentos com diferentes técnicas de mineração de dados, objetivando encontrar padrões entre as conformações do receptor que indicassem quais eram as mais promissoras. Foram executados experimentos de mineração de dados com técnicas como classificação, regressão e agrupamento, que permitiram um melhor entendimento da interação receptor-ligante e avançaram no sentido de reduzir o espaço conformacional a ser considerado nas simulações de docagem molecular com o modelo FFR.

1.2 Motivação

O processo de docagem considerando a flexibilidade explícita do receptor, a partir de uma série de execuções de simulações de docagem onde, em cada uma, uma conformação do receptor é considerada, é computacionalmente muito custoso. Por outro lado, a consideração da flexibilidade de receptores é muito importante, uma vez que proteínas não permanecem rígidas em seu ambiente celular, sendo a flexibilidade essencial para exercer sua função.

Assim, a principal motivação do trabalho está em contribuir para a redução do conjunto de conformações a serem consideradas, agrupando-as por diferentes critérios de similaridade, mas ainda mantendo as características explícitas de flexibilidade do receptor tornando a busca de novos inibidores mais realista e abrangente, já que a flexibilidade do receptor estará sendo considerada.

A utilização de mineração de dados em Bioinformática é uma linha de pesquisa em evidência atualmente. As técnicas de preparação de dados biológicos desenvolvidas neste trabalho poderão ser utilizadas em outras áreas de pesquisa em Bioinformática que necessitem da aplicação de técnicas para a descoberta de conhecimento em grandes bancos de dados.

Outra motivação para o desenvolvimento deste trabalho está relacionada ao receptor investigado, a enzima InhA do *Mycobacterium tuberculosis*. A tuberculose é uma doença infecciosa que, embora curável, representa um problema de saúde pública de proporção mundial. Dados da Organização Mundial da Saúde (OMS) reportam que 9,2 milhões de pessoas no mundo desenvolveram tuberculose no ano de 2006, com um total de 1,7 milhões de mortes relacionadas a esta doença [ORG08]. A isoniazida (INH), um dos principais fármacos utilizados no tratamento da tuberculose [ROZ98] tem como alvo a enzima InhA. Este fármaco apresenta uma poderosa atividade antibactericida, porém, devido ao tratamento longo, prescrição imprópria do medicamento, e muitas vezes falta de colaboração do paciente, surgiram cepas de *Mycobacterium tuberculosis* resistentes a um ou mais fármacos hoje existentes no mercado.

Pelos motivos apresentados, torna-se essencial a busca de inibidores alternativos para essa enzima. E para isso, há a necessidade da utilização e desenvolvimento de novas técnicas que agilizem o processo de Planejamento Racional de Fármacos (do inglês, *Rational Drug Design* - RDD).

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral dessa Tese é de contribuir para o entendimento da importância da flexibilidade do receptor em simulações de docagem molecular e para a redução do tempo necessário para a execução desse tipo de experimento a partir da aplicação de um processo de descoberta de conhecimento em Banco de Dados.

1.3.2 Objetivos Específicos

- Desenvolver um Banco de Dados para armazenamento de resultados de docagem molecular com o modelo FFR e preparar os dados para os diferentes algoritmos de mineração.
- Executar experimentos de mineração de dados com as técnicas de classificação e regressão de forma a melhorar o entendimento sobre a importância da flexibilidade de receptores em simulações de docagem molecular.
- Agrupar conformações do modelo FFR obtidas de uma simulação pela DM. Para tal, novas funções de similaridade são desenvolvidas e utilizadas por diferentes algoritmos de agrupamento de conformações.
- Contribuir para a redução do tempo de execução das simulações de docagem molecular com o modelo FFR com o auxílio do padrão de dados para workflows científicos P-MIA [HÜB10].

1.4 Organização da Tese

Esta Tese de Doutorado está organizada da seguinte forma:

- O Capítulo 2 apresenta os principais conceitos importantes para o entendimento de todo o trabalho. Neste capítulo são descritos: o Planejamento Racional de Fármacos ou Rational Drug Design (RDD), incluindo as suas principais etapas, o processo de Docagem Molecular, as diferentes abordagens para a consideração da Flexibilidade do Receptor em docagem molecular e a metodologia de Dinâmica Molecular.
- No Capítulo 3 são descritos todos os materiais e métodos utilizados. Nele inclui-se a descrição das principais ferramentas direta e indiretamente aplicadas: o *workflow* científico FReDoWS [MAC07], o software de docagem molecular AutoDock3.0.5 [GOO96], o software de dinâmica molecular AMBER9 [PEA95], o software para análise de interação receptor-ligante LigPlot [WAL95], o sistema gerenciador de banco de dados PostGreSQL [STO86], a linguagem de programação *Python* e a plataforma para mineração de dados WEKA [WIT05]. Além das ferramentas, esse capítulo descreve o receptor e os ligantes investigados, assim como, as simulações por DM e docagem molecular que originaram todos os dados desta Tese.

- O Capítulo 4 é uma continuação do anterior. Nele são abordados os conceitos de Mineração de Dados e das principais etapas do processo de descoberta de conhecimento em banco de dados. São também descritas neste capítulo as técnicas de mineração de dados utilizadas e os respectivos algoritmos.

- No Capítulo 5 é apresentado o primeiro resultado desta Tese, o Banco de dados FReDD (*Flexible Receptor Docking Database*), desenvolvido para armazenar os resultados de docagem molecular com o modelo FFR, assim como as informações sobre as conformações do receptor e os ligantes. Este capítulo também apresenta como o conteúdo armazenado no FReDD foi preparado para ser utilizado com as técnicas de mineração de dados onde é descrito o algoritmo desenvolvido para gerar essas entradas. Ao final desse capítulo, a partir das entradas preparadas para mineração é descrita uma análise preliminar sobre esses dados.

- O Capítulo 6 apresenta o segundo conjunto de resultados desta Tese, a aplicação da técnica de mineração de dados Classificação com árvores de decisão utilizando o algoritmo J48. Uma das principais contribuições desse capítulo é a metodologia proposta de discretização do atributo alvo dos arquivos de entrada utilizados. Essa metodologia proposta é então comparada com 2 métodos de discretização clássicos com base no impacto dos mesmos no resultado das árvores de decisão geradas.

- O Capítulo 7 descreve o terceiro conjunto de resultados desta Tese que estão relacionados com os experimentos realizados com a técnica de mineração de dados de regressão por árvores modelo, utilizando o algoritmo M5P do WEKA. Nele são comparados os resultados obtidos com árvores modelo para diferentes formas de pré-processamento dos arquivos de entrada. Além disso, é descrita uma metodologia de pós processamento dos resultados das árvores modelo que permitiu a indicação das conformações mais promissoras nesses experimentos.

- O Capítulo 8 contém o último conjunto de resultados desta Tese, que compreendem os experimentos com a técnica de mineração de dados Agrupamento. São apresentados os testes realizados com 10 algoritmos de agrupamento implementados em [SHA07] para diferentes entradas e com diferentes funções de similaridade, incluindo as funções desenvolvidas nesta Tese. No final deste Capítulo são descritas análises utilizando o P-MIA [HÜB10] que comparam as funções de similaridade e mostra um estudo de caso efetivo do ganho de processamento obtido com a utilização do P-MIA em conjunto com os resultados de Agrupamento.

- No Capítulo 9 relaciona alguns trabalhos já publicados com o conteúdo desta Tese. Estes incluem trabalhos sobre Bancos de Dados para Planejamento Racional de Fármacos, trabalhos sobre a execução de docagem molecular com o receptor flexível e seleção de conformações e trabalhos sobre a utilização de algoritmos de agrupamento com dados de DM.

- O Capítulo 10 apresenta as considerações finais desta Tese, com sugestões para trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta conceitos importantes para o entendimento de todo o trabalho. A primeira seção descreve o Planejamento Racional de Fármacos [KUN92] ou *Rational Drug Design* (RDD), incluindo as suas principais etapas. A seguir, a seção Docagem Molecular explica este processo que constitui o princípio do RDD. A seção Consideração da Flexibilidade do Receptor apresenta algumas das diferentes abordagens que podem ser adotadas para incorporar a flexibilidade do receptor em simulações de docagem molecular. Em seguida, é explicada brevemente a metodologia de Dinâmica Molecular. Esta metodologia para a geração de conformações de um receptor faz parte da abordagem de incorporação da flexibilidade do mesmo durante o processo de em docagem adotado neste trabalho. Por fim, são apresentadas as considerações finais deste capítulo.

2.1 O Planejamento Racional de Fármacos

A indústria farmacêutica tem investido constantemente em novas tecnologias para melhorar a qualidade dos compostos candidatos a fármacos [LYN02]. Paralelo a isso, os avanços da biologia molecular e de ferramentas de simulação *in-silico*, o planejamento de medicamentos passou a ser feito de maneira mais lógica, o que é chamado de Planejamento Racional de Fármacos [KUN92] (RDD). Esse processo consiste basicamente de quatro etapas descritas em [KUN92] e representadas no fluxograma da Figura 2.1:

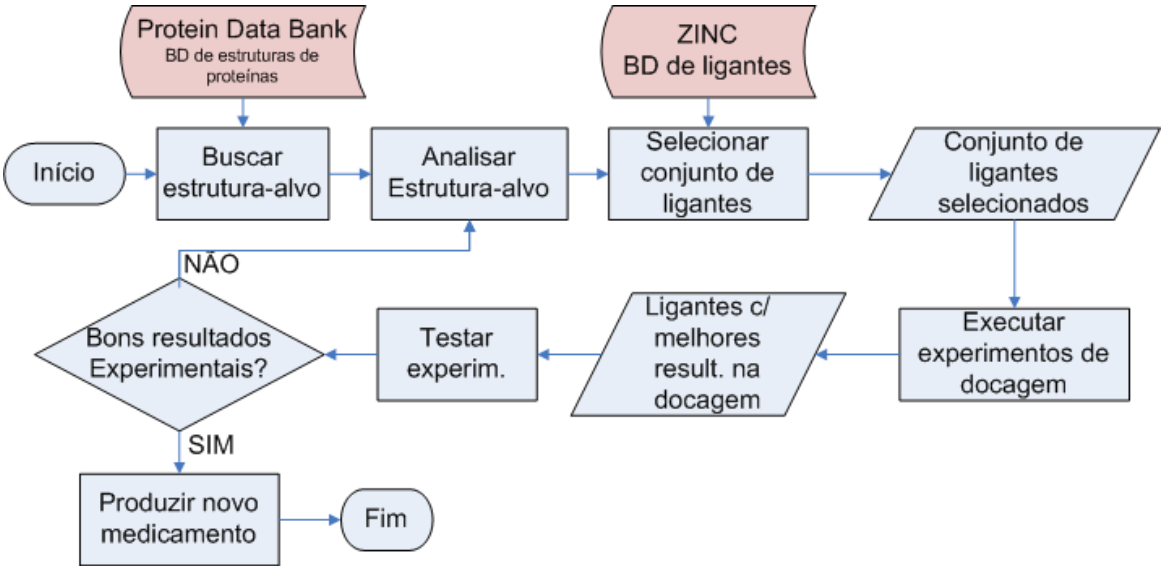


Figura 2.1: Fluxograma para representação do processo de planejamento de fármacos assistido por computador.

1. A primeira etapa consiste em isolar um alvo específico ou receptor (proteínas, receptores de membrana, DNA, RNA e outros). A partir da análise computacional da estrutura tridimensional (3D) dessa proteína determinada por modelagem comparativa ou cristalografia por difração

de raios-X ou por ressonância magnética nuclear, e armazenada em um banco de dados estrutural como o *Protein Data Bank* - PDB [BER00], é possível apontar prováveis regiões de ligação onde uma pequena molécula (ligante) pode se ligar a esse receptor (atividades 1 e 2 do fluxograma da Figura 2.1).

2. Baseado nas prováveis regiões de ligação identificadas na etapa anterior, é selecionado um conjunto de possíveis candidatos, chamados ligantes, que podem se ligar a essa região no receptor (atividade 3 do fluxograma da Figura 2.1). Usualmente ligantes podem ser buscados em bancos de dados de compostos como o ZINC [IRW05]. As diferentes conformações e orientações que determinado ligante pode assumir dentro do sítio de ligação de uma determinada proteína são simuladas por software de docagem molecular como AutoDock 3.0.5 [GOO96], FlexX [RAR96] e DOCK4.0 [EWI01] (esse passo está representado na atividade 4 do fluxograma descrito na Figura 2.1).

3. Os ligantes que teoricamente obtiveram melhores resultados nas simulações são experimentalmente sintetizados e avaliados através de ensaios biológicos e pré-clínicos (atividade 5 do fluxograma).

4. Baseado nos resultados experimentais, o medicamento é gerado (atividade 6 do fluxograma da Figura 2.1) ou o processo retorna ao início.

As quatro etapas do processo de RDD descritas por Kuntz [KUN92] compreendem a fase de pesquisa e desenvolvimento. Nesta fase do desenvolvimento de um novo medicamento, que é realizada geralmente entre 3 a 4 anos, são desenvolvidas as pesquisas *in silico* e *in vitro*. Segundo Silverman [SIL04], após essa fase de identificação, o novo medicamento passa por testes *in vivo*, divididos em duas etapas: testes pré-clínicos (onde são escolhidos os modelos animais para servir como cobaias) e testes clínicos em humanos. Os testes clínicos em humanos são divididos em 4 fases [SIL04]:

1. Fase I (3-18 meses): nesta fase é avaliada a segurança do novo fármaco, se os efeitos colaterais do mesmo são suportáveis, definir a melhor forma de administração do medicamento e analisar como o organismo reage. Os testes são aplicados em 20 a 100 voluntários saudáveis;

2. Fase II (1-3 anos): nesta fase é analisada a efetividade da droga. São determinados os efeitos colaterais e outros aspectos relacionados a segurança e toxicidade do composto. Durante esse período também é determinada a dosagem a ser administrada baseada em uma amostra de algumas centenas de pacientes;

3. Fase III (2-6 anos): estabelece a eficácia e efeitos colaterais após um longo período de uso do medicamento baseado em uma amostra maior de pacientes. O registro da nova droga é então submetido para o órgão responsável do país (por exemplos nos Estados Unidos, a US *Food and Drug Administration* - FDA) e após aprovado para comercialização ainda são necessários alguns meses para o acerto de questões burocráticas.

4. Fase IV: Consiste na pesquisa pós-comercialização, onde são realizados estudos da eficácia e segurança em uma parte da população doente. Nesta fase o medicamento já foi aprovado para ser comercializado.

2.2 Docagem Molecular

O entendimento detalhado das interações entre receptores biológicos e seus ligantes é muito importante para aplicações médicas e industriais, sendo essencial para a interpretação de uma série de fenômenos bioquímicos, constituindo a base do RDD [LEN96,LYB95]. A docagem molecular é um processo que possibilita que pequenas moléculas sejam posicionadas em uma configuração favorável para a formação de um complexo receptor-ligante estável. Esse método tem se mostrado muito efetivo no estudo de interações proteína-ligante e essa informação estrutural obtida do complexo modelado teoricamente pode auxiliar na definição de como a estrutura de um ligante pode ser modificada para melhorar sua função biológica ou para o desenho de novos compostos [HOU99].

A docagem molecular é executada por algoritmos de docagem, que são capazes de gerar um grande número de complexos receptor-ligante, avaliando-os em termos da energia livre de ligação (FEB - *Free Energy of Binding*), a qual quanto mais negativa, melhor a interação receptor-ligante. Na Figura 2.2 é apresentado um exemplo do processo de docagem molecular. Em cinza, está representado parte do sítio ativo da molécula receptora (a proteína InhA [DES95]). Em ciano a conformação inicial do ligante Triclosano antes da execução do algoritmo de docagem molecular, e em magenta, o ligante em sua posição final determinada pelo algoritmo de docagem como sendo a de melhor encaixe.

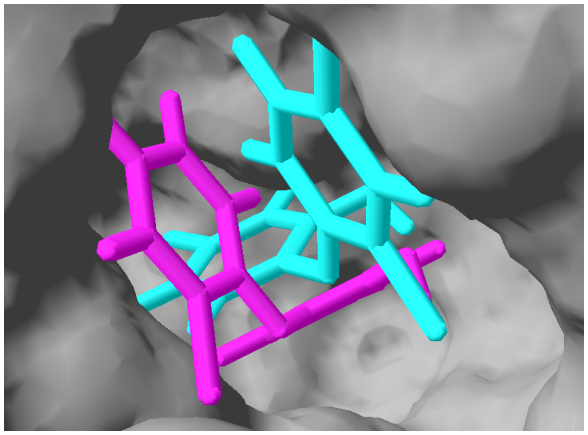


Figura 2.2: Representação esquemática do processo de docagem molecular em três dimensões (3D). A proteína é representada na forma de superfície, em cinza, e o ligante em palitos. A conformação inicial do ligante aparece em ciano, e a conformação do ligante ao final de um experimento de docagem molecular, em magenta.

O processo de analisar a interação receptor-ligante não é simples pois é influenciado por muitos fatores entrópicos e entálpicos, como por exemplo, a mobilidade do ligante e do receptor, o efeito do ambiente no receptor, a distribuição de cargas no ligante, e outras interações dos mesmos com

a água que dificultam muito a descrição desse processo. Dessa forma, independente da natureza dos complexos receptor-ligante, algumas questões precisam ser contempladas pelos algoritmos de docagem molecular e estas podem ser resumidas como a combinação da estratégia de busca com uma função de avaliação [SOT00]. A estratégia de busca da melhor conformação/orientação do ligante precisa explorar exaustivamente todas as formas de ligação entre ligante e receptor, o que inclui tanto a exploração de todos os seis graus de liberdade translacionais e rotacionais do ligante, quanto os graus de liberdade conformacionais do receptor.

Nos primeiros algoritmos de docagem molecular desenvolvidos, o método de aproximação mais comum era o tratamento de ligantes e receptores como corpos rígidos, baseado no modelo de reconhecimento molecular do tipo "chave e fechadura" proposto por Emil Fisher em 1894. Entretanto esta é uma simplificação muito drástica e limita os resultados para aqueles próximos à conformação experimentalmente observada para o complexo receptor-ligante [SCH04]. Além do mais, a geometria molecular pode mudar muito na associação receptor-ligante uma vez que ambas são moléculas flexíveis, sendo mais apropriada uma analogia ao modelo "mão-e-luva" uma vez que durante a docagem molecular ligante e receptor ajustam suas conformações de forma a encontrar um melhor encaixe, o chamado encaixe induzido ou *induced-fit*, proposto inicialmente por Koshland Jr. em 1958 [WEI04].

A flexibilidade do ligante é atualmente explorada pela maioria dos software de docagem molecular, uma vez que não envolve um grande esforço computacional. Entretanto, a consideração da flexibilidade de receptores continua sendo um grande desafio [TOT08]. Modelar diretamente os movimentos de uma proteína associado com a flexibilidade do sítio ativo da mesma representa um problema significante devido ao desafio duplo da alta dimensionalidade do espaço conformacional e a complexidade da função de energia envolvida [TOT08]. Esses fatores tornam a exploração de todos os graus de liberdade do receptor impraticável [WON08, COZ08].

Porém, ao mesmo tempo, não considerar a flexibilidade dos receptores nos experimentos de docagem molecular induz a uma predição errônea do local de ligação de 50 a 70% dos ligantes [TOT08]. Além do mais, proteínas não permanecem rígidas em seu ambiente celular e essa flexibilidade é essencial para exercerem suas funções.

2.3 Consideração da Flexibilidade do Receptor

Atualmente existe um grande número de alternativas para incorporar, ao menos, parte da flexibilidade do receptor revisadas nos últimos anos por Teodoro e Kavraki [TEO03], Totrov e Abagyan [TOT08], Cozzini *et al.* [COZ08], Wong [WON08], Alonso *et al.* [ALO06] e, mais recentemente, por B-Rao *et al.* [BRA09] e Yuriev *et al.* [YUR10]. Muitos avanços nos programas de docagem molecular publicados no ano de 2009 têm especificamente endereçado a questão da flexibilidade dos receptores [YUR10].

Os diferentes autores das revisões sobre o assunto classificam as abordagens de diferentes maneiras. Para Teodoro e Kavraki [TEO03] as abordagens podem ser agrupadas conforme mostra a Figura 2.3:

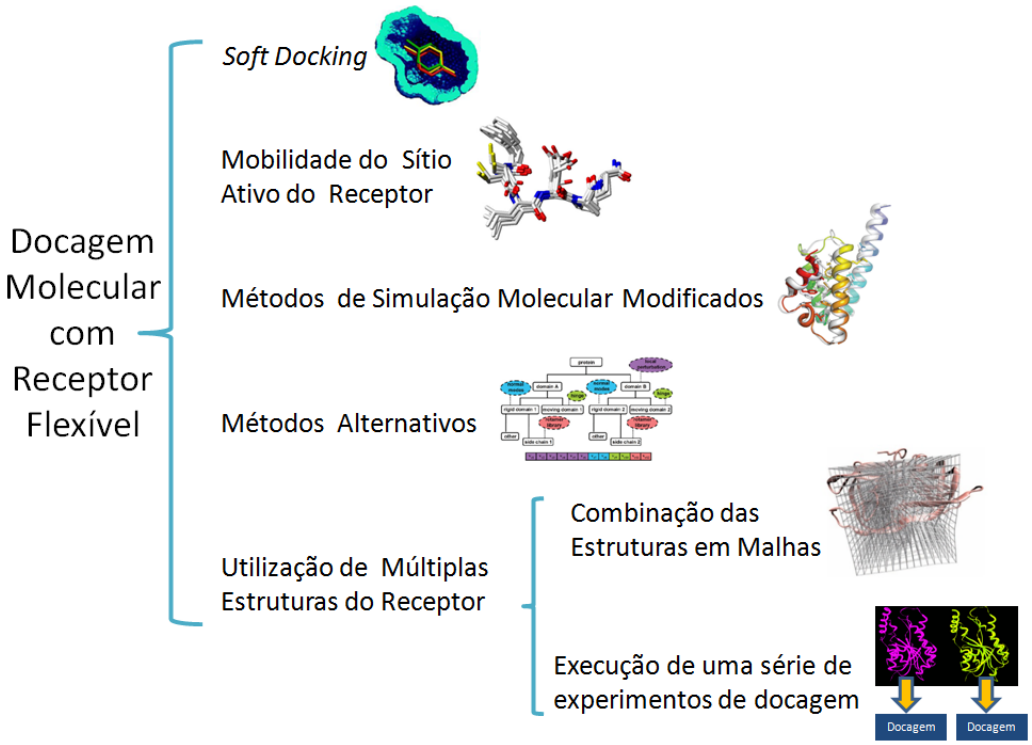


Figura 2.3: Representação esquemática das diferentes abordagens para incorporação da flexibilidade do receptor em simulações de docagem molecular de acordo com [TEO03].

2.3.1 *Soft Docking*

Segundo Alonso *et al.* [ALO06], a primeira solução chamada *Soft Docking* lida com a flexibilidade de maneira simples e indireta. Nessa abordagem, um pequeno grau de sobreposição entre o ligante e o receptor é permitido através de interações de van der Walls suavizadas [JIA91]. Isso permite, por exemplo, que um determinado ligante possa se encaixar em um sítio de ligação onde supostamente só uma molécula menor poderia. Segundo B-Rao *et al.* [BRA09] a maior desvantagem dessa abordagem é que somente movimentos nas cadeias laterais podem ser analisados mas não na cadeia principal ou outras mudanças mais significativas. Entretanto, *Soft Docking* tem a vantagem de ser computacionalmente eficiente e de fácil interpretação, sendo por isso também explorado em trabalhos como o de Ferrari *et al.* [FER04].

2.3.2 Mobilidade do Sítio Ativo do Receptor

Segundo Wong [WON08] o primeiro método a incorporar explicitamente a flexibilidade do receptor em simulações de docagem foi o método apresentado por Leach [LEA94]. Nessa abordagem, durante a docagem molecular são explorados os graus de liberdade conformacionais das cadeias laterais de alguns resíduos do sítio ativo do receptor, enquanto sua cadeia principal é mantida rígida. Uma extensão desse trabalho é apresentada em [LEA98] onde o espaço conformacional de toda a proteína é explorado.

Assim como a abordagem apresentada por Leach [LEA94], muitas outras metodologias de incor-

poração da flexibilidade do receptor em docagem molecular se utilizam da inclusão da mobilidade das cadeias laterais do sítio ativo do receptor como, por exemplo: Schnecke e Kuhn [SCH00] apresentam o algoritmo SLIDE em que um fragmento do ligante é inicialmente posicionado, seguido pela adição de outros fragmentos. Após, conflitos entre o ligante e o receptor são resolvidos por rotações em partes do ligante e nas cadeias laterais do receptor; Shaffer e Verkhivker [SCH98] descrevem um algoritmo que utiliza uma biblioteca de rotâmeros para executar uma busca otimizada de todas as possibilidades de combinações de conformações das cadeias laterais do sítio ativo do receptor.

Apostolakis *et al.* [APO98] apresenta uma abordagem que utiliza uma conformação do receptor na qual um ligante é posicionado aleatoriamente em seu sítio ativo e a energia do complexo é minimizada para remover eventuais sobreposições. Isto é repetido 1000 vezes, gerando uma estrutura diferente do sítio ativo a cada execução. Os melhores resultados são submetidos a um refinamento da minimização de energia. Dessa forma, o conjunto de conformações do sítio ativo representa parte da flexibilidade do receptor.

Mais recentemente, o programa AutoDock4 [MOR09] passou a modelar completamente a flexibilidade de certas porções da proteína. Isto é realizado a partir de cadeias laterais do receptor selecionadas pelo usuário que serão tratadas como flexíveis durante a simulação de docagem molecular, utilizando os mesmos métodos aplicados pelo AutoDock4 para tratar a flexibilidade do ligante.

Outro trabalho mais recente, que incorpora a flexibilidade do receptor ao processo de docagem a partir da mobilidade do sítio ativo do receptor, é o algoritmo MADAMM [CER09]. Em MADAMM, a proteína é flexibilizada utilizando-se uma biblioteca de rotâmeros de cadeias laterais de aminoácidos. MADAMM aumenta a capacidade do algoritmo de docagem introduzindo essa flexibilidade das cadeias laterais selecionadas pelo usuário através de um processo de docagem com múltiplos estágios incluindo uma etapa de modelagem molecular. Neste trabalho foi demonstrado que a orientação de resíduos particulares do receptor tem uma influência crucial na forma como receptor-ligante interagem durante a docagem molecular. Segundo B-Rao et al. [BRA09], a maior desvantagem desse método é que o mesmo somente considera mudanças conformacionais nas cadeias laterais.

2.3.3 Métodos de Simulação Molecular Modificados

Para simular o processo de ligação receptor-ligante o mais detalhado possível e evitar limitações de outras abordagens de modelo de flexibilidade, algumas metodologias se utilizam dos métodos de simulação pela DM, muitas vezes, em etapas pós-docagem [TEO03]. A principal vantagem da representação por DM em estudos de docagem é que essa se mostra muito acurada e pode explicitamente modelar todos os graus de liberdade do receptor. Infelizmente essas metodologias apresentam custos computacionais muito altos [TEO03]. O detalhamento deste tipo de técnica está fora do escopo deste trabalho.

2.3.4 Formas Alternativas de Representar a Flexibilidade do Receptor

Nos artigos de revisão [TEO03, TOT08, COZ08, WON08, ALO06, BRA09] além das técnicas classificadas em alguma das categorias de acordo com Teodoro e Kavraki [TEO03] há diversas metodologias que não se enquadram em nenhum dos outros 4 grupos.

Um exemplo é a técnica inovadora apresentada por Zhao e Sanner [ZHA07], o software FLIPDock (*Flexible Ligand Protein Docking*). Esse software permite a execução automática de docagem molecular considerando ligantes e sítios ativos do receptor como flexíveis descrevendo os mesmos por meio de *Flexibility Tree* (FT), uma estrutura de dados que codifica o subespaço conformacional de moléculas biológicas utilizando um pequeno número de variáveis e reduzindo muito o custo computacional de modelar moléculas flexíveis. A FT é utilizada para descrever tanto ligantes quanto receptores. Segundo Zhao e Sanner [ZHA07] a maior vantagem do FLIPDock é sua versatilidade para a descrição com a FT das moléculas envolvidas e a combinação com métodos de busca e como as funções de escore. Assim é possível que novas funções possam ser incorporadas a qualquer momento ao programa. O FLIPDock também permite ao usuário controlar a alocação de recursos computacionais para a representação de movimentos específicos.

2.3.5 Utilização de Múltiplas Estruturas do Receptor

Além dos métodos apresentados acima, que utilizam somente uma conformação do receptor, há um grande número de abordagens para incorporação da flexibilidade do receptor na docagem molecular que utilizam um conjunto de conformações do mesmo. Esse conjunto de conformações pode ser determinado experimentalmente por difração de raios X ou por NMR ou gerados por métodos computacionais como simulações utilizando DM.

2.3.5.1 Combinação de Múltiplas Estruturas do Receptor em *Grids*

Muitos trabalhos focam na descrição da flexibilidade do receptor através da combinação de um conjunto de estruturas em *grids*. Geralmente esses *grids* representam uma média das estruturas. Knegtel *et al.* [KNE97] foram os pioneiros nessa abordagem na qual condensaram as estruturas do receptor em um grid simples com o objetivo de reduzir o tempo de execução da docagem molecular. Eles avaliaram duas diferentes maneiras de combinar muitas estruturas determinadas experimentalmente em uma representação média: uma considerando a média de energia de interação entre receptor-ligante e outro baseado na variação de posição dos átomos do receptor. Osterberg *et al.* [OST02] oferecem uma representação discreta do sítio ativo do receptor através do cálculo de *grids* de três maneiras diferentes: um *grid* médio que corresponde a uma média simples ponto-a-ponto dos valores de todos os *grids* que representam as estruturas; um *grid* mínimo que considera o valor mínimo entre todos os *grids* e por último um *grid* ponderado pela energia envolvida, quanto mais negativa a energia, maior o peso do ponto no *grid* médio. Broughton [BRO00] também utiliza *grids* para representar a flexibilidade do receptor, entretanto as estruturas são obtidas de uma simulação pela DM.

Ao invés de combinar diferentes estruturas em um *grid* simples, o programa FlexE [CLA01] cria uma descrição única para a proteína alvo. As características estruturais mais conservadas são sobrepostas em uma estrutura média rígida. Para as regiões que variam, são consideradas explicitamente diferentes estruturas mantidas como um conjunto que é explorado combinatoriamente durante a docagem.

Mais recentemente, Bottegoni et al. [BOT09] têm trabalhado no desenvolvimento do *4D-Docking*, um novo protocolo para execução de docagem molecular, em que a conformação do receptor é a quarta dimensão. Neste protocolo, múltiplos *grids* representam múltiplas conformações e cada uma destas é considerada como uma variável na otimização global. Essa abordagem se mostrou bastante eficiente para a modelagem da flexibilidade de receptores em docagem molecular, reduzindo o tempo de execução desse tipo de experimento e mantendo a acurácia.

2.3.5.2 Execução de uma Série de Simulações de Docagem Utilizando Diferentes Estruturas do Receptor

Segundo Alonso *et al.* [ALO06], a abordagem mais abrangente para a inclusão da flexibilidade de receptores consiste em executar simulações de docagem molecular do ligante contra cada estrutura de um conjunto de estruturas do receptor geradas por simulação pela DM. Lin *et al.* [LIN02, LIN03] apresentam o método chamado RCS - *Relaxed Complex Scheme* para acomodar a flexibilidade do receptor na busca pela conformação receptor-ligante mais correta. Primeiro os autores executaram uma simulação pela DM do receptor sem o ligante e então docaram ligantes às estruturas geradas durante a simulação pela DM. Após, aplicaram o RCS para encontrar a conformação receptor-ligante mais correta. Mais recentemente, no trabalho apresentado por Amaro *et al.* [AMA08], os autores apresentam melhorias no método RCS incluindo uma redução prévia do conjunto de conformações do receptor, buscando um conjunto menor, porém representativo (os trabalhos de [LIN02, LIN03, AMA08] serão melhor detalhados no Capítulo de Trabalhos Relacionados).

Entre todas as metodologias que foram brevemente apresentadas neste trabalho, optamos por considerar a flexibilidade do receptor utilizando a combinação de docagem molecular com resultados de simulação por DM. Como explicado na Introdução desta Tese, foi adotado o termo modelo FFR (*Fully-Flexible Receptor*) para essa metodologia. Este consiste na execução de uma sequência de simulações de docagem molecular, em cada uma é empregada uma conformação diferente da trajetória da DM. Apesar dessa metodologia aumentar a chance de se encontrar um receptor em um estado conformacional correto para acomodar um ligante em particular [ALO06], ela tem como maior desvantagem o tempo necessário para executar cada diferente simulação de docagem receptor-ligante [TOT08, COZ08, WON08, HUA06].

2.4 Dinâmica Molecular

Em condições fisiológicas, as biomoléculas experimentam vários tipos de movimentos e de mudanças conformacionais muitas vezes cruciais para suas funções. Com o avanço de técnicas experi-

mentais tornou-se possível uma visão mais detalhada de diversos processos biológicos pelo acesso a propriedades atômicas de macromoléculas biológicas, como proteínas [van90]. O acesso a esse tipo de informação permitiu o desenvolvimento de estudos de simulação por DM que tem por objetivo simular o comportamento microscópico de átomos em proteínas, fundamentada nos princípios básicos da mecânica clássica. Com simulações pela DM é possível estudar o efeito explícito de ligantes na estrutura e estabilidade das proteínas, considerando os efeitos do solvente e os diferentes parâmetros termodinâmicos envolvidos (pressão, temperatura, volume, etc.), incluindo energias de interação e entropias.

Dessa forma, a simulação por DM é uma das técnicas computacionais mais amplamente aplicada no estudo de macromoléculas biológicas [van90, KAR00]. Essa técnica é importante para o entendimento do comportamento dinâmico das proteínas em diferentes intervalos de tempo, o que permite o estudo desde movimentos internos rápidos até mudanças conformacionais mais lentas [COH09]. Por essas razões, atualmente a simulação por DM é considerada a melhor técnica para obter um conjunto mais completo de conformações de uma proteína.

De acordo com Cozzini *et al.* [COZ08] a técnica de simulação pela DM representa o método mais acessível e com um custo mais razoável para a geração de conformações de um receptor. Por esse motivo, a metodologia de incorporação da flexibilidade do receptor em simulações de docagem molecular considerada neste trabalho se utiliza de conformações geradas por simulação pela DM.

2.5 Considerações Finais

Este capítulo apresentou os principais conceitos envolvidos no trabalho e que são essenciais para o seu entendimento: planejamento de fármacos e suas principais etapas, a docagem molecular e a simulação por DM. Também está descrito neste capítulo as diferentes abordagens que podem ser utilizadas para a consideração da flexibilidade de receptores em simulações de docagem molecular. Entre todas a metodologias que foram brevemente apresentadas, a que escolhemos neste trabalho foi a combinação de docagem molecular com resultados de simulação por DM. Para isso, é executada uma seqüência de simulações de docagem molecular onde em cada uma é empregada uma diferente conformação da trajetória da DM (modelo FFR).

A execução automática dessa série de simulações de docagem para o modelo FFR do receptor é feita utilizando-se o workflow científico FReDoWS descrito em [MAC07, MAC11a]. De acordo com Alonso et al. [ALO06] essa metodologia aumenta a chance de se encontrar um receptor em um estado conformacional correto para acomodar um ligante em particular. Entretanto, o tempo necessário para executá-la ainda é considerável, sendo esta a principal desvantagem desta abordagem [TOT08, COZ08, WON08, HUA06].

Outros trabalhos desenvolvidos no LABIO (Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas) utilizaram do modelo FFR, como os trabalhos [SCH04, SCH05, COH11, COH09], demonstrando a importância desse tipo de abordagem. No trabalho descrito em [SCH04, SCH05], a utilização das estruturas instantâneas geradas nas simulações por DM permitiu a simulação da

flexibilidade da enzima InhA de *Mycobacterium tuberculosis* [DES95] durante a docagem molecular no estudo da afinidade pelo NADH desta proteína e 3 diferentes mutantes da mesma (I21V, I16T e S94A). Neste trabalho foi demonstrado que as mutações causam instabilidades conformacionais ao longo das trajetórias dinâmicas, sendo este um conhecimento importante para a busca de novos inibidores para esta proteína. Essas informações não teriam sido obtidas sem considerar o receptor e seus mutantes como moléculas flexíveis.

No trabalho descrito por Cohen et al. em [COH09, COH11], foi investigado o efeito da flexibilidade explícita também da enzima InhA através da realização de simulações de docagem molecular em cada uma das diferentes conformações do seu modelo FFR (tipo selvagem e mutantes I16T e I21V), com os inibidores etionamida (ETH), triclosan (TCL) e isoniazida-pentacianoferrato II (PIF). O modelo FFR utilizado neste estudo mostrou que diferentes modos de ligação dos ligantes ETH, TCL e PIF não poderiam ser avaliados se somente uma conformação da enzima InhA tivesse sido utilizada. A análise apresentada nestes trabalhos [COH09, COH11] revelou, por exemplo, que para o complexo InhA-ETH apenas 5 resíduos da proteína interagem com este ligante na estrutura cristalina, enquanto que ao longo da trajetória no seu modelo FFR, 80 diferentes resíduos fazem contatos com o ligante ETH. O mesmo também foi observado nos estudos com o ligante TCL, onde apenas 2 resíduos deste receptor interagem com o ligante na estrutura cristalina, enquanto que no modelo FFR, 46 diferentes resíduos interagem com o TCL. Efeito semelhante é observado para o complexo InhA-PIF e para os resultados de docagem com os mutantes I16T e I21V e os mesmos ligantes. Isto indica que quando a plasticidade do receptor é considerada em simulações de docagem molecular é permitido que sejam explorados novos espaços no sítio de ligação do receptor, que não seriam possíveis de outra forma [COH09].

Além desses trabalhos descritos brevemente acima, um estudo detalhado da importância do modelo FFR na docagem esta descrito em [WIN10a] e será detalhado nos próximos capítulos. Neste trabalho foi analisado os resultados da docagem molecular também do modelo FFR da InhA espécie selvagem, com 4 ligantes: NADH, ETH, PIF e TCL. Foi demonstrado que, por exemplo para o ligante NADH, em sua estrutura cristalográfica, 22 resíduos do receptor interagem com o ligante, enquanto que utilizando o modelo flexível deste, 185 diferentes resíduos interagem (e neste trabalho interagir significa que o resíduo esteve a uma distância máxima de 4,0 $\mathrm{\AA}$ do ligante em algum momento). A mesma diferença é observada para os outros ligantes, demonstrando novamente a importância da consideração da flexibilidade do receptor, pois com o uso de um modelo rígido não seria possível a obtenção desse tipo de informação, que é muito importante na busca de novos inibidores para esta enzima.

O próximo capítulo apresenta os materiais e métodos utilizados neste trabalho. Serão detalhadas todas as ferramentas aplicadas no desenvolvimento desta Tese. Este próximo capítulo também inclui a descrição do receptor e dos ligantes utilizados, assim como das simulações de docagem molecular que originaram todos os dados.

3. MATERIAIS E MÉTODOS

Neste capítulo são descritos todos os materiais e métodos utilizados no desenvolvimento deste trabalho. A primeira seção descreve todas as ferramentas direta e indiretamente aplicadas no presente trabalho, que incluem: o *workflow* científico FReDoWS [MAC07] desenvolvido no LABIO e utilizado para execução de simulações de docagem molecular com o receptor flexível, o software de docagem molecular AutoDock3.0.5 [GOO96], o software de dinâmica molecular AMBER9 [PEA95], o software para análise de interação receptor-ligante LigPlot [WAL95], o sistema gerenciador de banco de dados PostGreSQL [STO86], a linguagem de programação *Python* e a plataforma para mineração de dados WEKA [WIT05]. As seções seguintes apresentam a descrição do receptor e ligantes utilizados assim como as simulações por dinâmica molecular e docagem molecular que originaram todos os dados utilizados nesta Tese. Por fim são feitas as considerações finais deste capítulo.

Os dados descritos neste capítulo e utilizados como material na Tese estão, em parte, em artigos publicados durante o desenvolvimento deste trabalho:

- o artigo completo publicado no LNBI-LNCS [MAC07] durante o evento *Brazilian Symposium on Bioinformatics* de 2007;
- os resumos publicados e artigos submetidos ao X-meeting 2010 e em avaliação (2° rodada) para publicação no *BMC Bioinformatics* [COH11, MAC11a];
- o resumo publicado e apresentado durante o evento ISCB-Latin America [COH10] em 2010;

3.1 Ferramentas utilizadas no desenvolvimento deste trabalho

3.1.1 FReDoWS

Para a execução automática de simulações de docagem molecular, considerando o modelo FFR foi desenvolvido um *workflow* científico, o qual denominamos de FReDoWS [MAC06, MAC07, MAC11a] (*Flexible-Receptor Docking Workflow System*). O artigo descrito em [MAC11a] é uma extensão significativa do trabalho introdutório apresentado em [MAC07] e que inclui uma etapa de seleção de conformações do receptor a serem utilizadas nas simulações de docagem molecular. A Figura 3.1 mostra o modelo final do FReDoWS utilizado neste trabalho para a execução das simulações de docagem molecular.

No modelo do FReDoWS (Figura 3.1) cada tipo de atividade do workflow corresponde a uma cor diferente. As atividades em verde-escuro são executadas pelo usuário, as atividades em roxo representam *subflows*, sub-processos compostos por atividades, transições e aplicações próprias. As atividades em verde-claro são executadas pelo sistema sem intervenção do usuário e podem invocar uma ou mais aplicações externas. As atividades em rosa são utilizadas para sincronização no modelo e para transações com condições mais complexas, nenhuma ação é efetivamente executada por elas.

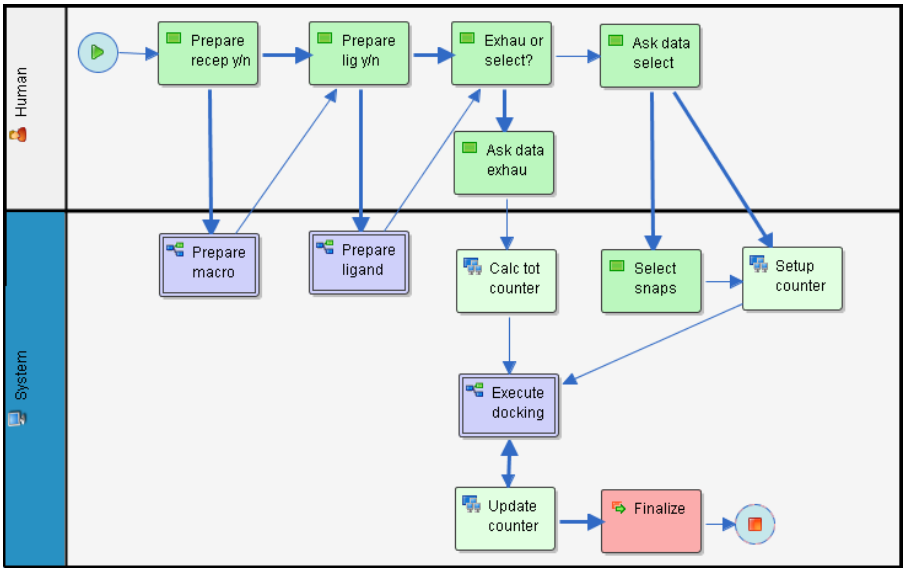


Figura 3.1: Modelo Final do *workflow* FReDoWS [MAC11a]. Detalhes do modelo no texto.

As principais etapas da execução do FReDoWS são resumidas a seguir [MAC06,MAC07,MAC11a]:

1. preparação dos arquivos da macromolécula. Nesta etapa, a DM já foi executada e os arquivos do receptor são preparados para uso na docagem molecular. No caso dos programas utilizados pelo FReDoWS, essa etapa inclui a execução do módulo Ptraj do software AMBER (que será detalhado a seguir) para transformar os arquivos resultados da DM em arquivos PDB;

2. preparação do ligante: Nessa etapa o ligante é posicionado onde se deseja que seja sua conformação inicial nas simulações de docagem molecular;

3. o usuário seleciona o tipo de experimento, se é *Exaustivo* ou *Seletivo*. No experimento *Exaustivo*, todas as conformações do modelo FFR são utilizadas. No experimento *Seletivo* é estabelecido um critério de seleção de conformações, onde o total a ser utilizado na docagem é estabelecido pelo usuário. Essa etapa de seleção será detalhada no Capítulo de Trabalhos Relacionados;

4. independente de ser *Exaustiva* ou *Seletiva*, a execução dos experimentos de docagem é realizada da mesma forma, pelo *subflow Execute Docking*. Como para isso utilizamos o programa AutoDock3.0.5, essa etapa de execução do *workflow* será descrita na próxima seção.

3.1.2 AutoDock3.0.5

O AutoDock3.0.5, programa de docagem molecular empregado neste trabalho, consiste em um conjunto de programas de código e acesso livres (Addsol, AutoTors, AutoGrid e AutoDock) desenvolvidos por Olson et al. [GOO96, MOR98] para a predição do modo de ligação de ligantes com receptores macromoleculares. Ele combina um método baseado em malhas (do inglês, *grid*) para a avaliação da energia do complexo, pré-calculando as energias de interação receptor-ligante

par-a-par e utilizando estas para otimizar o cálculo da energia final a cada iteração [MOR98]. As principais etapas envolvidas na execução de docagem molecular com o AutoDock3.0.5 são:

1. Preparação dos arquivos do receptor e do ligante. Para a preparação das cargas atômicas parciais do receptor, considerando que o mesmo já é um arquivo .PDBQ (arquivo .PDB com cargas), é executado o módulo do AutoDock chamado *addsol*. O *addsol* especifica parâmetros de solvatação atômica para cada átomo da macromolécula, gerando um arquivo .PDBQS [MOR01]. A preparação do ligante, considerando que o mesmo já esteja em um formato .MOL2, compreende a execução do módulo *deftors*, onde podem ser definidos os ângulos de torção do ligante.

2. A segunda etapa consiste na execução dos módulos *mkgpf3* e *mkdpf3*, responsáveis por gerar os arquivos de parâmetros para os módulos *Autogrid* e *Autodock* respectivamente.

3. É executado o módulo Autogrid. Esse módulo define uma malha de afinidade para cada um dos tipos de átomos do ligante: tipicamente carbono, oxigênio, nitrogênio, enxofre e hidrogênio, mais uma malha de potencial eletrostático. Essa malha corresponde a uma matriz 3D de pontos igualmente espaçados, centrado em alguma região de interesse do receptor em estudo (Figura 3.2(a)).

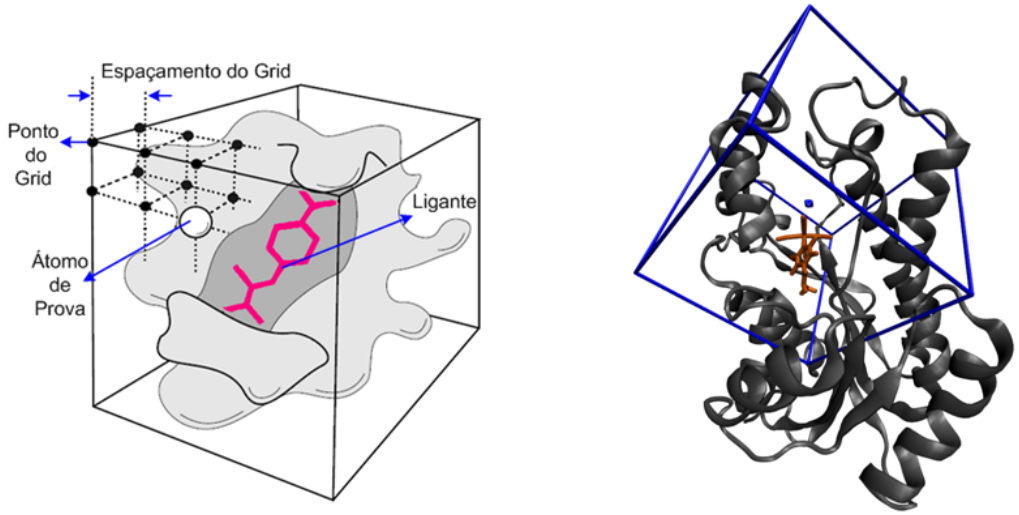


Figura 3.2: (a) Exemplo da malha de afinidade gerada pelo Autogrid. Adaptada de [MOR01]. (b) Exemplo de uma malha de afinidade em um receptor. Nesta figura trata-se do receptor InhA, onde a malha está centrada no ligante PIF definida com os valores padrão do Autogrid, 60x60x60 Å.

   Cada ponto dentro da malha armazena a energia potencial de um átomo de prova em relação a todos os átomos no receptor. Assim, durante a docagem esses valores de energia são utilizados para reduzir os cálculos que são necessários para se chegar ao valor final da FEB.

4. Na última etapa é então executado o módulo Autodock. Para a execução do Autodock é escolhido um dos 3 algoritmos descritos a seguir. Adicionalmente, independente da função

de busca utilizada para encontrar o melhor encaixe receptor-ligante, é calculada a FEB do complexo conforme detalhado em [MOR98].

Algoritmo SA (*Simulated Anneling*): Na exploração conformacional utilizando a técnica de Monte Carlo com SA, a proteína permanece estática durante a simulação e a molécula do ligante faz um movimento aleatório dentro do sítio de ligação. A cada passo da simulação é aplicada uma modificação pequena e aleatória em cada grau de liberdade do ligante. Esta modificação resulta numa nova configuração, cuja energia é avaliada utilizando a malha de afinidade previamente calculada pelo AutoGrid. Esta nova energia é comparada com a energia da etapa anterior. Se a nova energia for menor, a nova configuração é imediatamente aceita. Se a nova energia for maior, o resultado é tratado probabilisticamente em função da temperatura [GOO96].

Algoritmo GA (*Genetic Algorithm*): Na aplicação de GA para a docagem molecular, uma certa posição do ligante no sítio ativo do receptor (definida por um conjunto de valores que descrevem a translação, orientação e conformação deste), determinam seu estado. Cada variável de estado do ligante corresponde a um gene. O estado do ligante corresponde ao genótipo, onde suas coordenadas atômicas são seu fenótipo. O GA do AutoDock3.0.5, então inicia sua execução pela criação aleatória de uma população com um número de indivíduos definidos pelo usuário. A criação da população inicial é seguida de ciclos sucessivos de gerações até que o número máximo de gerações ou o número máximo de avaliações de energia seja alcançado. A avaliação de energia é calculada baseada em uma função de energia resultante da interação receptor-ligante. Durante esse processo, pares aleatórios de indivíduos passam por um processo de *crossover* (recombinação), no qual os novos indivíduos herdam os genes dos pais e/ou mutação randômica, na qual os genes sofrem uma alteração aleatória [MOR98].

Algoritmo LGA (*Lamarckian genetic algorithm*: A partir do AutoDock 3.0, o algoritmo genético Lamarckiano passou a estar disponível para uso. O LGA é um método híbrido que combina um algoritmo de busca global (GA) com um algoritmo de busca local (do inglês, *Local Search* - LS). No LS são aplicadas pequenas alterações rotacionais e conformacionais, atuando no genótipo. No LGA as adaptações provenientes do LS são como adaptações em função do ambiente, sendo então possíveis de serem transferidas para as próximas gerações se apresentarem uma melhor avaliação de energia [MOR98].

No AutoDock é executado um determinado número de *runs*, que correspondem ao número de diferente tentativas que serão executados pelo algoritmo selecionado, de forma a encontrar a melhor energia de interação entre o receptor-ligante (FEB). Dessa forma, como resultado da execução de uma simulação de docagem utilizando o Autodock3.0.5 tem-se um arquivo de saída que lista as coordenadas do ligante, FEB, RMSD (com relação a posição inicial) e outros valores para cada uma das tentativas executadas (*runs*), ordenadas ascendentemente por FEB.

3.1.3 AMBER9

O Amber é uma coleção de programas que permitem aos seus usuários a execução de simulações por DM de biomoléculas [PEA95]. Para a execução de uma simulação por DM com o AMBER são necessários: as coordenadas cartesianas de todos os átomos do sistema, a topologia (determina a conectividade entre os átomos, seus nomes, etc.), o campo de força e a lista de comandos, que determinarão os parâmetros da simulação. O Amber não é um programa de acesso livre e código aberto. Entretanto, alguns de seus módulos são, como por exemplo, seus campos de força, o módulo *Ptraj*, etc.

A DM que deu origem as conformações utilizadas neste trabalho foi executada com o AMBER6 [CAS99]. Os arquivos resultantes dessa DM foram processados com o módulo *Ptraj* do AMBER9 [CAS06], que transforma esses arquivos em .PDBs. Nesta versão do *Ptraj*, foram incluídos algoritmos de agrupamento de estruturas e os mesmos foram aplicados neste trabalho, conforme será detalhado no Capítulo 8.

3.1.4 LigPlot

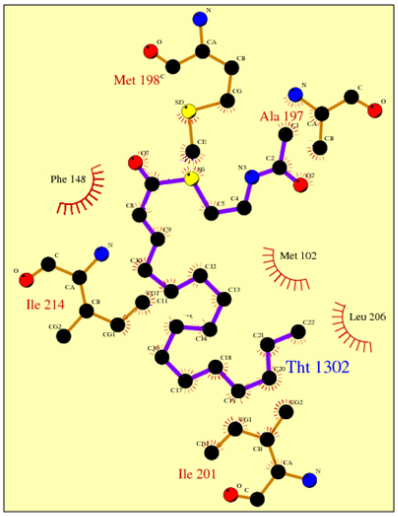
O LigPlot [WAL95] é um programa que gera esquemas da interação entre um receptor e um ligante a partir de um arquivo PDB. Sua saída é um arquivo PostScript (ps) que contém uma representação das interações do complexo, incluindo ligações de hidrogênio e contatos hidrofóbicos. Em [WAL95] é descrito o algoritmo deste programa. O LigPlot, assim como o AutoDock3.0.5 é um programa de código e acesso livre.

O LigPlot se utiliza dos contatos calculados previamente pelo programa HBPLUS [McD94]. Então, como entrada para a execução do LigPlot são fornecidos: um arquivo .PDB único do receptor e ligante, os arquivos gerados pelo HBPLUS (arquivo .hhb que contém a lista de ligações de hidrogênio, o arquivo .nnb com a lista de contatos hidrofóbicos e um arquivo de configuração com os parâmetros para a plotagem). Como saída, os principais arquivos gerados pelo LigPlot são (Figura 3.3):

1. LigPlot.ps - Figura 3.3(a). Nesse arquivo, as ligações de hidrogênio são indicadas por linhas tracejadas entre os átomos envolvidos. Os contatos hidrofóbicos são marcados nos resíduos por um desenho de um arco com traços ao redor, em direção ao átomo do ligante que faz esse contato [WAL95]. No exemplo apresentado na figura foi executado o LigPlot considerando como entrada um arquivo .PDB com um receptor, a enzima InhA [DES95] e com um ligante, o substrato THT presente na estrutura cristalográfica desta enzima (Código PDB: 1ENY).

2. LigPlot.pdb: contém o .PDB que está no arquivo .ps (contém o ligante e os resíduos do receptor que estabelecem contato com o mesmo);

3. LigPlot.hhb - Figura 3.3(b): arquivos no formato LigPlot que contém as ligações de hidrogênio do .hhb original. Em Figura 3.3(b) nota-se que há somente uma ligação de hidrogênio encontrada e a mesma é entre átomos do próprio THT (por essa razão não aparece no ligplot.ps);



(a)

ligplot.hhb output:

Donor			Acceptor			Distance
THT	1302	N3	THT	1302	O2	2.25

(b)

ligplot.nnb output:

Atom 1			Atom 2			Distance
THT	1302	C16	ILE	214	CD1	3.08
THT	1302	C13	ILE	214	CD1	2.82
THT	1302	C12	ILE	214	CD1	3.07
THT	1302	C15	ILE	214	CG1	2.61
THT	1302	C14	ILE	214	CG1	3.04
THT	1302	C20	LEU	206	CD2	3.30

(c)

1eny_tht

Interacting residues	Hydrogen bonds M-M	S-S	Non-bonded M-S	contacts
Residues: [THT 1302] -> [ILE 214]	0	0	0	5
Residues: [THT 1302] -> [LEU 206]	0	0	0	1
Residues: [THT 1302] -> [ILE 201]	0	0	0	14
Residues: [THT 1302] -> [MET 198]	0	0	0	7
Residues: [THT 1302] -> [ALA 197]	0	0	0	4
Residues: [THT 1302] -> [PHE 148]	0	0	0	6
Residues: [THT 1302] -> [MET 102]	0	0	0	1

(d)

Figura 3.3: Arquivos de saída do Programa LigPlot executado considerando um .PDB do complexo InhA-THT. (a) LigPlot.ps contém uma representação visual dos contatos receptor-ligante. (b) LigPlot.hhb apresenta as ligações de hidrogênio. (c) LigPlot.nnb lista os contatos hidrofóbicos determinados pelo HBPLUS. (d) LigPlot.sum, sumariza as ligações de hidrogênio e contatos hidrofóbicos em um arquivo, listando o total de contatos feito por cada resíduo do receptor envolvido nos mesmos.

4. LigPlot.nnb - Figura 3.3(c): arquivos no formato LigPlot que contém os contatos hidrofóbicos do .nnb. A Figura 3.3(c) mostra parte desse arquivo, onde se pode ver por exemplo, que o THT estabelece 5 contatos com o resíduo do receptor ILE (Isoleucina) 214 (na última coluna desse arquivo é listada a distância entre os átomos do contato). Esses contatos podem ser verificados na Figura 3.3(a);

5. LigPlot.sum: é um arquivo que sumariza as informações dos arquivos .hhb e .nnb, conforme mostra a Figura 3.3(d). Neste arquivo estão listados somente os totais de contato de cada resíduo do receptor, sem mostrar qual é exatamente o átomo que estabeleceu o contato e a respectiva distância do mesmo.

O LigPlot foi aplicado neste trabalho para a determinação dos contatos entre o receptor e seu substrato, o que foi utilizado como entrada para auxiliar no agrupamento das estruturas obtidas da DM, conforme será detalhado no Capítulo 8.

3.1.5 SGBD PostGreSQL

O PostgreSQL é um sistema gerenciador de banco de dados (SGBD) baseado no POSTGRES desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley [STO86]. O PostgreSQL é desenvolvido em código fonte aberto, com acesso livre, e suporta grande parte do padrão SQL além de funcionalidades como: comandos complexos, chaves estrangeiras, gatilhos, visões, etc. O Banco de Dados FReDD, que será descrito no Capítulo 5, foi desenvolvido utilizando esse SGBD.

3.1.6 Linguagem de Programação *Python*

A linguagem de programação *Python* foi adotada neste trabalho para o desenvolvimento de todos os *scripts* para tratamento dos dados, tanto para inserir dados no banco de dados desenvolvido, quanto para gerar as entradas para os algoritmos de mineração. Também foi utilizada na escrita de programas para o processamento dos resultados obtidos, em diferentes etapas do desenvolvimento desta Tese.

Python é uma linguagem de programação de alto nível, interpretada, imperativa e de tipagem dinâmica e forte. É simples de aprender, orientada a objetos e que contém estruturas de dados de alto nível. Por suas características, é a linguagem ideal para o desenvolvimento de aplicações rápidas para diferentes áreas e plataformas [POS11]. Para o desenvolvimento de alguns *scripts*, foi utilizada a biblioteca Biopython [COC09]. Essa biblioteca é composta por um conjunto de funcionalidades para biologia computacional, como por exemplo suporte a dados de diferentes formatos (FASTA, GenBank, saída do BLAST, etc.), interface com o BLAST, ClustalW, entre outras.

3.1.7 WEKA

O *Waikato Environment for Knowlodge Analysis* - WEKA [WIT05, HAL09] foi criado com a tarefa de unir diferentes algoritmos de aprendizagem de máquina em uma plataforma única. No início dos anos 90, esses algoritmos eram feitos para uso em diferentes plataformas e operavam com uma variedade de formatos de dados [HAL09]. Sendo assim, o WEKA, escrito em código aberto, fornece um conjunto de algoritmos de aprendizagem, e permite também que pesquisadores desenvolvam novos com o apoio de uma infra-estrutura para manipulação dos dados e os incluam neste ambiente. Atualmente, o WEKA tem ampla aceitação nos meios acadêmicos e empresariais e foi utilizado durante todo o desenvolvimento desta Tese para a execução dos experimentos de mineração de dados (detalhados nos próximos capítulos).

3.2 Receptor Investigado: Proteína InhA de *Mycobacterium tuberculosis*

A proteína em estudo é a InhA [DES95] (código PDB: 1ENY), a enzima 2-*trans*-*enoil ACP(CoA) Redutase* de *Mycobacterium tuberculosis* (Mtb) cuja estrutura 3D está apresentada na Figura 3.4. A InhA caracteriza-se por uma folha de 7 fitas-β paralelas, contornadas por 8 hélices-α, conectadas por alças e voltas, formando o sítio de ligação desta enzima com a coenzima NADH (nicotinamida adenina dinucleotídeo, forma reduzida), em azul na Figura 3.4.

A InhA é uma enzima importante no mecanismo de ação da tuberculose [OLI07] pois é responsável pela biossíntese de ácidos graxos, um importante componente da parede celular da micobactéria, e, consequentemente, uma das estruturas essenciais para a sua sobrevivência. Por esse motivo, desperta atenção especial como alvo atraente para o desenvolvimento de novos fármacos para a tuberculose [AGU08].

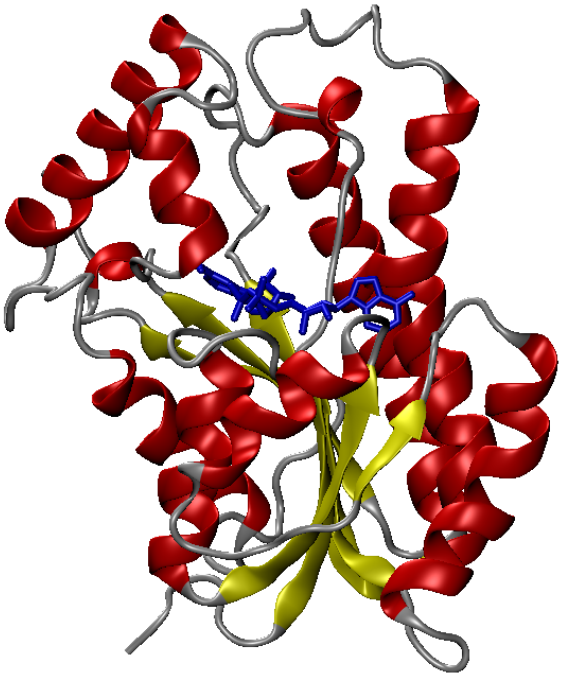


Figura 3.4: Estrutura do tipo *ribbons* 3D da proteína InhA na forma NewCartoon (software VMD). Em vermelho as hélices. Em amarelo as fitas. Em cinza as voltas e alças. Em azul, na forma Licorice (software VMD), a coenzima NADH.

3.3 Ligantes Considerados: NADH, PIF, TCL e ETH

Neste trabalho foram considerados quatro ligantes: NADH, PIF, TCL e ETH. O NADH - Nicotinamida Adenina Dinucleotídeo, forma reduzida [DES95] é a coenzima da proteína InhA. Essa molécula tem um total de 71 átomos (após a preparação para a docagem molecular, permanece com um total de 52 átomos pois perde os hidrogênios apolares) e sua estrutura 3D está descrita na Figura 3.5. Esse ligante foi considerado rígido em todos as simulações de docagem, por isso não será explorado seus possíveis ângulos de torção. Ele foi considerado dessa forma porque considerá-lo flexível aumentava o tempo de execução de cada simulação de docagem e os resultados em termos de FEB eram muito parecidos.

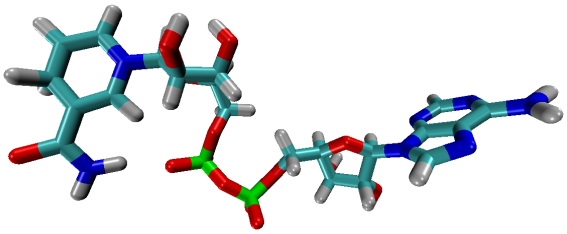


Figura 3.5: Estrutura 3D do ligante NADH. Os átomos da Figura 3.5 (e também das Figuras 3.8, 3.6, 3.7) estão coloridos da seguinte forma: Hidrogênio em cinza, Oxigênio em vermelho, Nitrogênio em azul, Carbono em ciano, Enxofre em amarelo, Fósforo em verde, Ferro em laranja e Cloro em magenta.

O ligante Isoniazida Pentacianoferrato (IPF) [OLI04] - Figura 3.6 é composto por 28 átomos antes da preparação do ligante para a docagem molecular e 24 átomos, após. Neste trabalho chamaremos esse ligante por sua sigla em inglês, PIF, *Pentacyano(isoniazid)ferrate* II. O PIF é uma molécula inibidora da InhA, desenvolvida por Oliveira e colaboradores [OLI04] para ser um inibidor sem ativação prévia. Consiste na molécula de Isoniazida (INH) acrescido de um grupamento pentacianoferrato com o centro metálico acoplado. Para a docagem com esse ligante flexível foram selecionados três ângulos de rotação, entre os átomos N3 e Fe, entre C8 e C13 e entre N14 e N16 (ligações em verde na Figura 3.6).

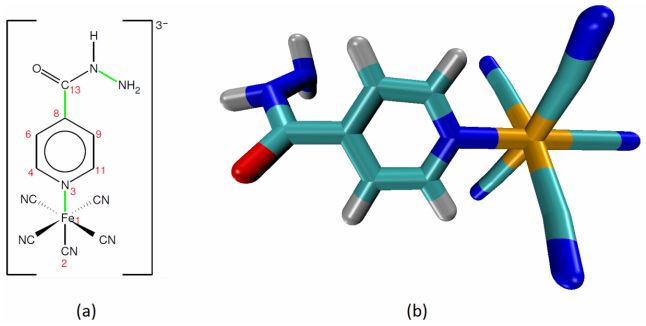


Figura 3.6: Estrutura 3D do ligante PIF. Adaptada de [COH09].

O ligante Triclosano (TCL) (código Zinc: 2216) [KUO03], Figura 3.7, é uma molécula pequena, formada por 24 átomos (antes da preparação para a docagem molecular, após, o ligante tem 18 átomos). De acordo com o banco de dados Zinc [IRW05], a molécula apresenta dois ângulos de rotação, o primeiro entre os átomos de carbono C3 e O2 e o segundo entre os átomos O2 e C7 [COH09] (ligações em verde na Figura 3.7). O AutoDock detecta um terceiro ângulo de rotação entre os átomos C4 e O1 mas este não é tratado quando este ligante é considerado como flexível nas simulações de docagem, sendo considerados somente os outros 2 ângulos mencionados.

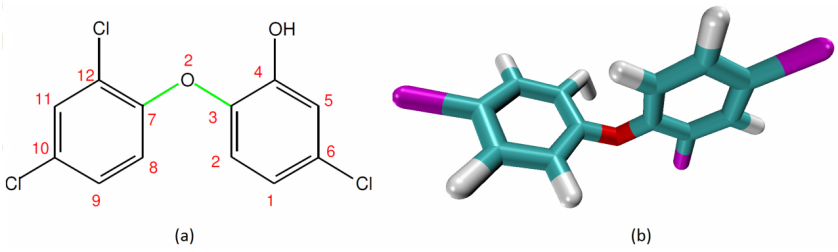


Figura 3.7: Estrutura 3D do ligante TCL. Adaptada de [COH09].

A Etionamida ou ETH (código Zinc: 4476370) [WAN07] é uma pequena molécula composta por 21 átomos (13 átomos após a preparação para a docagem molecular) e descrita na Figura 3.8. Este é um análogo estrutural da INH, amplamente utilizado no tratamento da tuberculose. Assim como a INH, a ETH também é uma pró-droga que necessita de ativação prévia inibindo a atividade da InhA quando covalentemente ligada ao NADH, formando um aduto NADH-ETH [BAU00]. De acordo com o banco de dados Zinc, essa molécula apresenta dois ângulos de rotação: o primeiro entre os átomos de carbono C2 e C3 e o segundo entre os átomos C5 e C8 (ligações em verde na

Figura 3.8). Apesar do AutoDock3.0.5 detectar um terceiro ângulo de rotação entre os átomos C8 e N2 [COH09], somente esses 2 primeiros serão considerados nas simulações de docagem molecular com a ETH flexível.

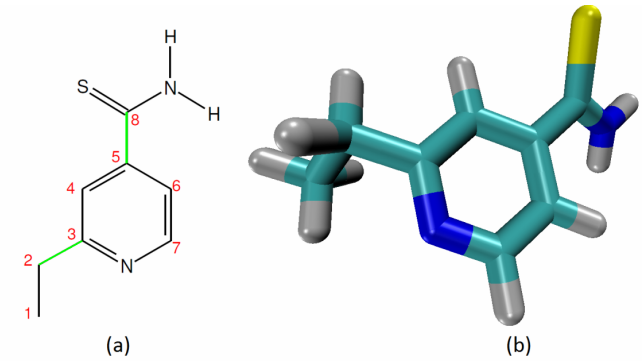


Figura 3.8: Estrutura 3D do ligante ETH. Adaptada de [COH09].

3.4 Simulações pela DM do Receptor InhA

Os estudos de simulação por DM da InhA de Mtb, que originaram as conformações utilizadas nesse trabalho, foram realizados com a InhA complexada com a coenzima NADH utilizando o software AMBER6.0 [CAS99] por um período de 3.100 ps (1 ps $= 10^{-12}$ segundos) e estão descritos no trabalho de Schroeder et al. [SCH04, SCH05]. Um exemplo da flexibilidade dessa proteína está na Figura 3.9, onde em (a) a flexibilidade é evidenciada pelas cadeias laterais de 3 conformações diferentes, a cristalográfica e as estruturas nos instantes 1.000 e 2.000 ps da simulação e em (b) tem-se diferentes conformações da InhA, onde cada cor também representa a conformação em um instante diferente no tempo.

3.5 Simulações de Docagem molecular

Durante o desenvolvimento desta Tese foram executados 2 conjuntos de simulações de docagem molecular, denominados Experimentos-Fase 1 e Experimentos-Fase 2. Os resultados dos Experimentos Fase 1 foram os utilizados durante a maior parte deste estudo. Porém, ao executar a última parte deste trabalho foi necessária uma modificação na forma de extrair as conformações da DM, o que causou uma modificação nas mesmas. Por esse motivo, as simulações de docagem precisaram ser reexecutadas, conforme descrito na Subseção a seguir.

3.5.1 Experimentos Fase 1

Para cada um dos 4 ligantes NADH, ETH, PIF e TCL foram submetidos 3.100 simulações de docagem molecular (ou seja, considerou-se todas as conformações da InhA geradas durante a simulação pela DM) utilizando o FReDoWS [MAC07, COH11, MAC11a, COH10]. A tabela 3.1 resume os resultados obtidos durante a execução das simulações de docagem molecular utilizando

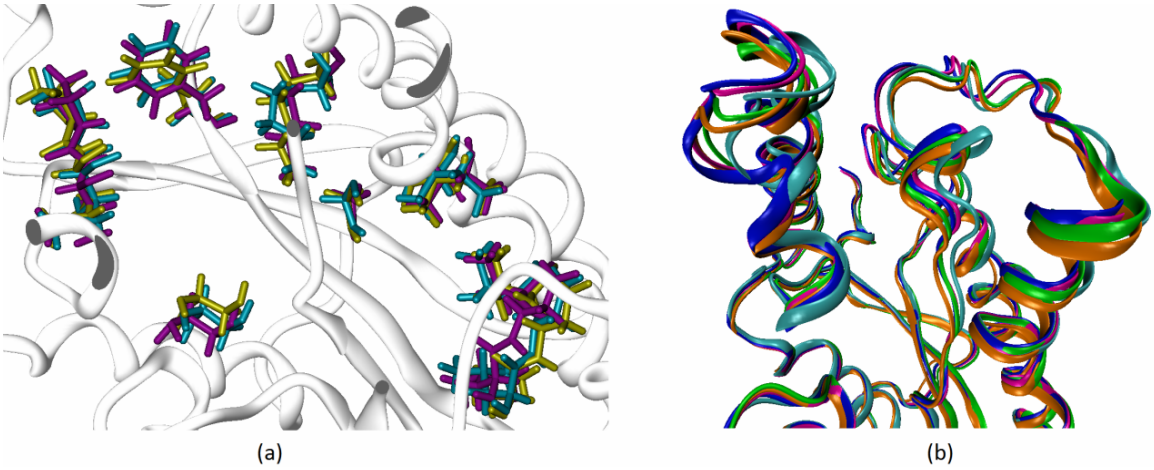


Figura 3.9: Exemplo da flexibilidade da proteína InhA em diferentes momentos ao longo de uma simulação por DM. (a) Em cinza parte da cadeia principal da estrutura cristalográfica da proteína. Em magenta, ciano e amarelo 10 cadeias laterais de aminoácidos da proteína cristalográfica e nos instantes de tempo 1.000, 2.000 ps da simulação por DM, respectivamente. (b) Em laranja a estrutura cristalográfica, em ciano a estrutura média de 0 a 500 ps, em azul de 550 a 1.000 ps, em magenta de 1.050 a 1.500 ps e em verde e 1.550 a 2.000 ps.

o AutoDock3.0.5 [GOO96] e como protocolo de execução o algoritmo SA, com parâmetros padrão e 10 *runs* onde os ligantes foram mantidos rígidos.

Tabela 3.1: Resultados das simulações de docagem molecular Fase 1. Detalhes no texto.

Ligantes	Média de FEB(-) *kcal/mol*	Total Resul. Válidos	Mínima FEB *kcal/mol*	Máxima FEB *kcal/mol*	Moda FEB *kcal/mol*
NADH todos os *runs*	-9,2 ± 4,5	11.284	-20,6	0	-16,2
NADH *run* de melhor FEB	-12,9 ± 4,2	2.823	-20,6	0	-16,8
PIF todos os *runs*	-9,1 ± 1,6	30.420	-11,2	0	-9,8
PIF *run* de melhor FEB	-9,9 ± 0,6	3.042	-11,22	0	-9,9
TCL todos os *runs*	-8,2 ± 1,3	28.370	-10,0	-0,7	-8,8
TCL *run* de melhor FEB	-8,9 ± 0,3	2.837	-10,0	-4,9	-9,0
ETH todos os *runs*	-6,4 ± 0,3	30.430	-8,2	-5,2	-6,6
ETH *run* de melhor FEB	-6,8 ± 0,3	3.043	-8,2	-5,9	-6,7

Na tabela 3.1, as linhas 1, 3, 5 e 7 correspondem aos resultados considerando os 10 *runs* de execução da docagem para os ligantes NADH, PIF, TCL e ETH respectivamente. As linhas 2, 4, 6 e 8 mostram os resultados considerando somente o *run* de melhor FEB durante a execução da docagem de cada arquivo de saída do AutoDock3.0.5 para os ligantes NADH, PIF, TCL e ETH respectivamente. A primeira coluna da tabela descreve o tipo de resultado (se é considerando todos os *runs* ou somente o de melhor FEB). Na segunda coluna está a média de FEB e o desvio padrão em *kcal/mol*, considerando somente os valores de FEB negativos. O total de simulações de docagem válidas descritos na terceira coluna corresponde ao total de simulações que convergiram para um valor de FEB negativo. Na quarta coluna é descrito o melhor valor de FEB para cada simulação

(FEB mínima). As colunas 5 e 6 contém os valores de FEB máximo e moda respectivamente. Os valores de FEB mínima, máxima e moda estão relacionados somente com os resultados de docagem válidos.

É importante ressaltar que o valor de FEB mínima para todas as simulações não se encaixa dentro do limite determinado pela média da FEB e desvio padrão. Isso ocorre porque a FEB apresenta em seu histograma, para ambos os ligantes, um comportamento bimodal, ou seja, há um valor de moda no qual a maioria dos demais valores de FEB se concentra, e um segundo valor de moda, onde algumas instâncias tem seu valor de FEB. Como a média foi calculada considerando todas as instâncias com valor de FEB negativas há essa diferença no seu desvio padrão que não inclui então o valor da melhor FEB.

Nesse conjunto de simulações os melhores resultados foram do ligante NADH, principalmente se somente o melhor *run* for considerado (média de FEB de -12,9 $\pm$ 4,2 Kcal/mol), sendo também para esse ligante a maior variação de FEB. O resultado para o ligante ETH apresentou a pior média de FEB (-6,4 Kcal/mol), porém ainda aceitável, conforme mostra a discussão apresentada em [MAC11a, MAC07, MAC06].

3.5.2 Experimentos Fase 2

Para a etapa final deste trabalho foram reexecutadas as simulações de docagem molecular para o mesmo receptor (e sua DM de 3.100 conformações) e os mesmos 4 ligantes. Essas simulações precisaram ser reexecutadas pois durante a utilização dos algoritmos de agrupamento (Capítulo 8), a DM precisou ser sobreposta na primeira estrutura (até o momento utilizávamos as estruturas no mesmo sistema de referência, mas não exatamente sobrepostas). Por esse motivo, as conformações foram alteradas em relação as utilizadas até o momento e armazenadas no banco de dados FReDD (descritos no Capítulo 3). Além do mais, havia a necessidade de avaliação dos resultados de docagem molecular com o ligante também flexível e com o algoritmo do AutoDock3.0.5 mais utilizado, o LGA.

As novas docagens também utilizaram o AutoDock3.0.5 [GOO96] e foram executados com o seguinte protocolo (descrito em detalhe em [COH11, MAC11a, COH10]):

- o algoritmo de execução do AutoDock3.0.5 selecionado foi o LGA;
- considerou-se 25 *runs* de execução com 500 mil avaliações em cada *run*;
- com exceção do NADH, os ligantes foram considerados flexíveis durante a execução do AutoDock3.0.5, sendo:
  - PIF com 3 ângulos de torção: entre os átomos N3_Fe, C8_C13 e N14_N16;
  - TCL com 2 ângulos de torção: entre os átomos C3_O2 e O2_C7 ;
  - ETH com 2 ângulos de torção: entre os átomos C2_C3 e C5_C8;

A Tabela 3.2 resume os resultados dos Experimentos Fase 2. A primeira coluna mostra o ligante; a segunda coluna contém a média e desvio padrão do valor de FEB para o resultado de melhor FEB,

a terceira coluna descreve o total de simulações válidas de cada ligante (aqueles cujo valor de FEB é negativo) e a última coluna apresenta o valor de FEB mínima para cada ligante.

Tabela 3.2: Resultados das simulações de docagem molecular Fase 2.

Ligantes	Média de FEB(-) *kcal/mol*	Total Resul. Válidos	Mínima FEB *kcal/mol*
NADH *run* de melhor FEB	-7,0 $\pm$ 2,6	2.770	-14,7
PIF *run* de melhor FEB	-9,7 $\pm$ 1,3	3.100	-13,7
TCL *run* de melhor FEB	-12,3 $\pm$ 0,5	3.100	-14,1
ETH *run* de melhor FEB	-9,6 $\pm$ 0,4	3.100	-10,8

Analisando a Tabela 3.2 é possível concluir que nesse segundo conjunto de simulações de docagem, os melhores resultados foram com o ligante TCL (média de FEB de -12,3 Kcal/mol), sendo os piores para o ligante NADH, que assim como para a Fase 1, na Fase 2 também apresenta maior variação de FEB (média de FEB de -7,0 $\pm$ 2,6 kcal/mol). Comparando as Tabelas 3.1 e 3.2 pode-se concluir que há diferenças entre os resultados, mesmo que o modelo FFR e ligantes sejam os mesmos para ambos experimentos. Essas diferenças ocorrem principalmente devido aos experimentos terem sido executados com algoritmos diferentes (SA e LGA), que conforme descrito na Seção 3.1.2 apresentam métodos de busca bem distintos e pelos ligantes serem tratados como flexíveis nos Experimentos Fase 2.

3.6 Considerações Finais

Este capítulo apresentou as principais ferramentas utilizadas no desenvolvimento desta Tese (FReDoWS, AutoDock3.0.5, AMBER9, LigPlot, PostGreSQL, *Python* e WEKA) assim como a descrição do receptor e ligantes considerados e as simulações por DM e docagem molecular que originaram todos os dados empregados. Conforme apresentado na seção Ferramentas, com exceção do software utilizado na geração da DM do receptor de estudo (AMBER), todas as demais ferramentas são software livre e com código aberto que executam no sistema operacional Linux.

No trabalho de Schroeder et al. [SCH05] foi demonstrada a flexibilidade do receptor InhA (Figura 3.9), que juntamente com a necessidade de desenvolvimento de novos fármacos para este importante alvo da Tuberculose, tornaram o estudo desse receptor interessante. Além do mais, outros trabalhos do LABIO [COH09, COH11, MAC11a, MAC07, MAC06] mostraram que o modelo FFR da InhA se reflete nos resultados de docagem molecular, conforme discutido no final do Capítulo 2, o que também justifica o uso deste tipo de receptor no presente trabalho.

No próximo capítulo serão apresentados os conceitos sobre a área de Mineração de Dados, descrevendo as técnicas de mineração aplicadas neste trabalho: classificação, regressão e agrupamento e os respectivos algoritmos utilizados.

4. MINERAÇÃO DE DADOS

Esse capítulo é uma continuação do anterior que descreve os Materiais e Métodos. Por compreender uma série de itens e para facilitar a leitura o seu conteúdo foi apresentado em separado. São abordados os conceitos de Mineração de Dados e das principais etapas do processo de descoberta de conhecimento em banco de dados. Além do mais, são descritas as técnicas de mineração de dados utilizadas, assim como, os algoritmos de cada técnica aplicados nesta Tese. Para finalizar esse capítulo, são apresentadas as considerações finais e uma breve descrição do capítulo seguinte.

Nessas considerações finais é apresentado o primeiro trabalho desenvolvido utilizando técnicas de mineração de dados aplicadas à dados de docagem molecular com o modelo FFR. Esse trabalho foi o ponto de partida de todos os resultados apresentados nos capítulos seguintes desta Tese, tendo sido publicado:

- como resumo na conferência X-meeting em 2007 [MAC08b] onde ganhou o prêmio de 3°melhor pôster do evento;
- como artigo completo no LNBI-LNCS [MAC08a] durante o evento *Brazilian Symposium on Bioinformatics* de 2008.

4.1 Descoberta de Conhecimento em Bancos de Dados e Mineração de Dados

A mineração de dados é uma parte integral da descoberta de conhecimento em bancos de dados (Knowledge Discovery in Databases) (KDD) [FAY96], que compreende o processo completo de converter dados crus em informação útil, como pode ser visto na Figura 4.1. O processo de KDD consiste de uma série de passos de transformação, desde o pré-processamento dos dados até o pós-processamento dos resultados da mineração de dados [TAN05] e compreende as seguintes etapas:

- O pré-processamento é a transformação dos dados de entrada em um formato apropriado para uma análise subsequente. Os passos envolvidos no pré-processamento incluem a integração dos dados de múltiplas fontes, a limpeza dos dados para remoção de ruídos e dados duplicados e a seleção de registros e características que serão relevantes na etapa de mineração de dados. Para Tan et al. [TAN05], o pré-processamento é a etapa mais trabalhosa e que ocupa mais tempo no processo de KDD.
- A etapa de mineração de dados cujas técnicas podem ser utilizadas para investigar grandes bancos de dados como o objetivo de encontrar padrões previamente desconhecidos, úteis e relacionados entre os dados. Um conjunto de fatores motivou o desenvolvimento de técnicas para a mineração de dados [TAN05], como, por exemplo, a necessidade diária de se lidar com bancos de dados de tamanhos na ordem de gigabytes ou terabytes que compreendem centenas

ou milhares de atributos, como os bancos de dados de expressão de genes, sendo necessárias técnicas para o tratamento dessa alta dimensionalidade. Outros problemas dizem respeito à complexidade e heterogeneidade dos dados, como os de estrutura sequencial e tridimensional do DNA ou proteínas, e aos dados distribuídos que precisam ser analisados mesmo não estando armazenados fisicamente em conjunto.

- O pós-processamento é a etapa que integra os resultados da mineração de dados a um sistema de suporte à decisão, permitindo que somente os resultados úteis sejam incorporados a esse sistema. De acordo com Tan et al. [TAN05], um exemplo dessa etapa consiste na visualização dos resultados da mineração de dados ou na aplicação de medições estatísticas para desconsiderar resultados não legítimos.

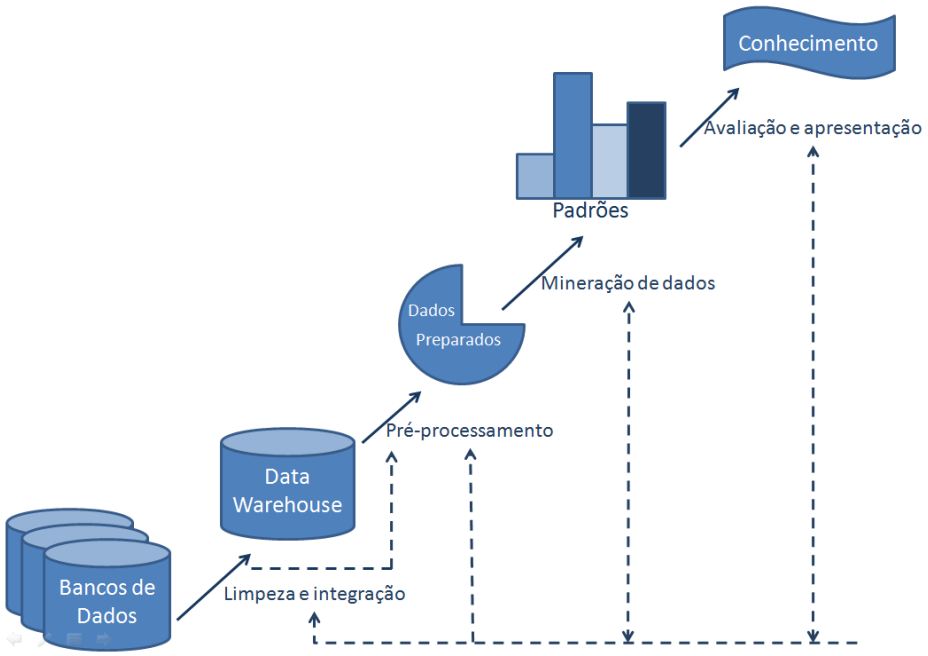


Figura 4.1: Mineração de dados como uma etapa do processo de descoberta de conhecimento. Adaptado de Han e Kamber [HAN06]

4.1.1 Mineração de Dados em Bioinformática

Atualmente, muitas pesquisas na área de mineração de dados estão voltadas para o desenvolvimento de ferramentas que lidam com diversos métodos escaláveis e eficientes para procurar padrões de interesse em grandes bancos de dados. Ao mesmo tempo em que o progresso na biologia e na ciência médica tem aumentado a necessidade de se lidar com o acúmulo de grandes quantidades de dados. A questão levantada por Han [HAN02] diz respeito a como integrar essas duas áreas: mineração de dados e Bioinformática, permitindo que a mineração em dados biológicos seja realizada com sucesso.

Métodos avançados, eficientes e escaláveis de mineração de dados ainda precisam ser desenvolvidos, e, dentro desse contexto, os seguintes tópicos são citados como possibilidades de pesquisa nessa área [HAN02]:

- procura por similaridade e comparação de dados biológicos: dados biológicos geralmente apresentam ruídos e dados faltantes, sendo importante o desenvolvimento de algoritmos de mineração que tratem esse tipo de problema;
- análise de associações: identificação de recorrência de sequências biológicas (de proteínas, por exemplo) ou outros padrões relacionados. Métodos de análise de associação e correlação podem auxiliar na determinação dos tipos de genes ou proteínas que podem ocorrer em amostras alvo. Essas análises facilitam o descobrimento de grupos de genes ou proteínas e o estudo da interação e relação entre eles;
- padrões frequentes em *grupos*: a maioria dos algoritmos que realizam agrupamento são baseados em distâncias Euclidianas ou densidade. Entretanto, dados biológicos geralmente consistem de vários atributos que formam um espaço de dimensões muito grande, sendo necessário descobrir padrões que permitam um correto agrupamento desses dados;
- análise de caminhos: unir informações sobre genes e proteínas em diferentes estágios da evolução de uma doença, pois os genes e proteínas influenciam cada um desses estágios de evolução. Se uma sequência de atividades dentro de cada estágio for estabelecida é possível desenvolver fármacos específicos com alvo nos diferentes estágios separadamente;
- visualização de dados e mineração de dados visual: estruturas complexas de sequências de genes e proteínas são mais bem representadas em grafos, árvores, cubos e cadeias por vários tipos de ferramentas de visualização. Essa forma de visualização dos dados facilita o entendimento de padrões, descobrimento de conhecimento e exploração de dados interativamente;
- preservação da privacidade dos dados biológicos: embora a troca de informações seja importante, hospitais e institutos de pesquisa permanecem relutantes em disponibilizar seus dados biológicos, sendo por isso importante desenvolver métodos de mineração de dados que preservem a privacidade dos dados.

4.2 Pré-processamento

Segundo Wang *et al.* [WAN04], dados biológicos são gerados, na maioria das vezes, em locais geograficamente diferentes, com uma variedade de recursos e pela aplicação de diferentes técnicas. Sendo assim, para extrair informações úteis, esses dados precisam ser agrupados, caracterizados e limpos. Essa etapa de pré-processamento dos dados pode consumir muito tempo se for necessário pesquisar muitos bancos de dados distribuídos para garantir a qualidade dos dados.

O principal objetivo do pré-processamento é garantir a qualidade dos dados e, de acordo com Tan *et al.* [TAN05], suas principais abordagens envolvem:

- Agregação: É a combinação de dois ou mais objetos em um objeto único permitindo que os conjuntos de dados sejam reduzidos, tornando o processamento dos mesmos mais rápido. O maior problema da agregação é uma possível perda de detalhes importantes. Em atributos quantitativos, corresponde a operações de soma ou média, e em atributos qualitativos, corresponde a valores que podem ser resumidos ou organizados em um conjunto.

- Amostragem: Seleciona um subconjunto do banco de dados, assim, ao utilizar somente esse subconjunto, são obtidos os mesmos resultados com um custo computacional reduzido.

- Redução de dimensionalidade: em Bioinformática, essa abordagem é muito importante, já que muitos bancos de dados biológicos apresentam centenas e até milhares de atributos. A principal vantagem em reduzir a dimensionalidade dos dados está no fato de que a maioria dos algoritmos de mineração de dados trabalha melhor se o número de atributos não for muito grande, o que permite a geração de modelos mais compreensíveis [TAN05].

- Seleção de um subconjunto de atributos: é outra maneira de reduzir a dimensionalidade dos dados. A eliminação de atributos redundantes (cujas informações já estão contidas em outros atributos) ou irrelevantes (dependendo da análise a ser realizada, alguns atributos não precisam ser considerados) torna as atividades de classificação ou agrupamento dos dados mais eficiente [TAN05]. Existem três abordagens para esse tipo de seleção dos dados: ou está inserida na etapa de mineração de dados, ou acontece antes como uma tarefa independente, ou é tratada como uma caixa preta dentro da etapa de mineração de dados.

- Criação de atributos: segundo Tan *et al.* [TAN05], muitas vezes é interessante criar, a partir dos atributos originais, um novo conjunto de atributos que capture informações importantes dos conjuntos de dados de forma mais eficaz. E, se esse número for menor que o original, haverá uma redução de dimensionalidade.

- Discretização e binarização: muitos algoritmos de mineração de dados, principalmente para determinação de padrões de associação entre os atributos precisam que os dados sejam binários, enquanto que algoritmos para classificação dos dados precisam que os atributos sejam categóricos, assim, essas transformações são muito utilizadas.

- Transformações de variáveis: refere-se a transformações aplicadas a todos os valores de uma variável, ou seja, para cada objeto, a transformação é aplicada ao valor da variável do objeto. Uma técnica muito utilizada de transformação é a normalização.

4.3 Técnicas de Mineração de Dados

Na Figura 4.2 é apresentado o banco de dados para o armazenamento de informações sobre a taxonomia de alguns indivíduos, como por exemplo, a ordem e família a que pertencem juntamente com algumas de suas características. A partir desse banco de dados, por meio da aplicação de

técnicas de mineração de dados, pode-se analisar sua árvore genealógica, agrupar indivíduos da mesma espécie, fazer uma predição da família a que pertencem com base nas suas características e antecedentes entre outras possibilidades, sendo esses somente alguns exemplos de informações que podem ser extraídas de bancos de dados a partir do uso de técnicas como associação, classificação e agrupamento.

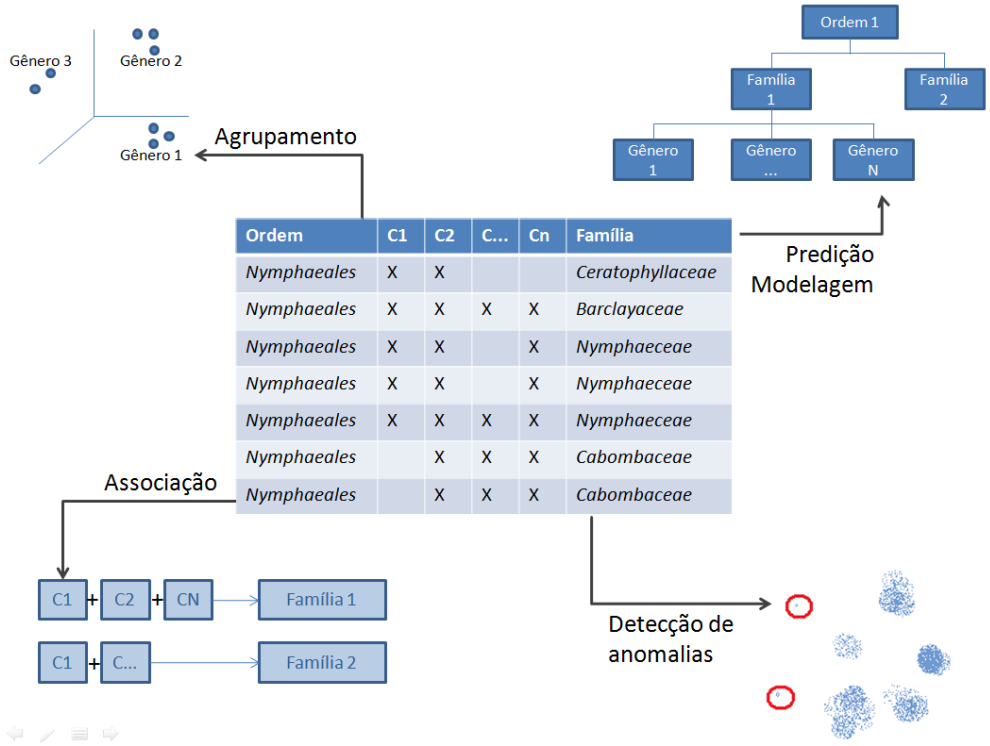


Figura 4.2: Exemplo de quatro das principais técnicas de mineração de dados no contexto de um banco de dados biológico.

Geralmente as tarefas de mineração de dados são divididas em duas categorias:

1. Técnicas Preditivas: O objetivo desse tipo de técnica é predizer um atributo baseado na análise dos valores de outros atributos. Existem dois tipos de modelos de predição: classificação e regressão.

2. Técnicas Descritivas: Têm por objetivo encontrar padrões entre os dados (correlações, grupos, anomalias, etc.) que explicam alguma relação entre os mesmos. Geralmente esse tipo de técnica necessita de pós-processamento para validar e explicar os resultados. As técnicas de mineração de dados descritivas são: associação, agrupamento e sumarização.

A seguir são apresentadas as técnicas de mineração de dados aplicadas neste trabalho: Classificação, Regressão, Agrupamento e Associação.

4.3.1 Classificação

Para Wang *et al.* [WAN04], dados biológicos consistem de múltiplos atributos e a relação/interação entre esses atributos pode ser muito complicada de ser estabelecida. Em Bioinformática,

classificação é uma das ferramentas mais populares e utilizadas para entender a relação entre características de vários objetos, pois trata-se de um processo que encontra propriedades comuns entre um conjunto de dados e os organiza em diferentes classes, de acordo com um modelo de classificação.

Para Tan *et al.* [TAN05], uma técnica de classificação, ou classificador, pode ser definida como uma abordagem sistemática para construir modelos de classificação a partir de dados de entrada onde aplica-se um algoritmo de aprendizado para induzir o modelo que melhor identifica as relações entre atributos. Os dados de entrada em uma tarefa de classificação são um conjunto de registros, ou instâncias, caracterizado por uma tupla (x, y), onde x é o conjunto de atributos e y o atributo-classe. A classificação consiste então na tarefa de aprender uma função-alvo f que mapeie cada conjunto de atributos x para um dos atributos-classe y, que deve ser categórico. Essa função-alvo f é um modelo de classificação.

Exemplos de abordagens de classificação incluem árvores de decisão, classificadores baseados em regras, redes neurais, redes Bayesianas, etc. Entre essas diferentes abordagens para classificação, nesse trabalho serão apresentados somente classificadores baseados em árvores de decisão.

4.3.1.1 Classificação Baseada em Árvores de Decisão

Segundo Chen *et al.* [CHE96], esse método de classificação é um método de aprendizado supervisionado que constrói árvores de decisão a partir de um conjunto de exemplos. A qualidade da árvore depende da exatidão da classificação e do tamanho da árvore. Cada árvore de decisão é composta por:

- nodo raiz: não apresenta arestas chegando e zero ou mais arestas saindo;
- nodos internos: cada um possui uma aresta chegando e duas ou mais saindo;
- folhas ou nodos terminais: cada nodo-folha possui uma aresta chegando e nenhuma saindo.

Esse tipo de modelo de aprendizagem, uma árvore de decisão, é composta por um nodo inicial, ou nodo raiz, a partir do qual vão sendo associados os nodos internos que contêm condições que testam os valores dos atributos, enquanto que os nodos-folhas estão associados a um determinado valor do atributo classificador (Figura 4.3). Sendo assim, a classificação de um registro de teste inicia pelo nodo raiz, onde aplica-se um teste relacionado ao valor do atributo associado ao nodo, dependendo desse valor, determina-se o próximo nodo a ser analisado. A partir desse outro nodo interno, para o qual uma nova condição é avaliada, é definido qual é o nodo no próximo passo, e assim acontece até se chegar a um nodo terminal ou folha, que definirá a que classe determinado registro pertence. Esses modelos seguem uma aproximação do tipo divisão-e-conquista uma vez que a medida que se vai percorrendo a árvore em direção as folhas, o número de possibilidades de resolução do problema vai diminuindo até se chegar à uma única solução, o que torna a classificação correta. No exemplo da Figura 4.3 é apresentada uma árvore de decisão para indicar qual é a classe de FEB (atributo classe: Bom, Regular ou Ruim) que determinado experimento de docagem

molecular apresenta depende da distância de alguns resíduos do receptor para o ligante (atributos preditivos). Então, percorrendo essa árvore de decisão pode-se verificar, por exemplo, que se a distância do resíduo do receptor HIE 92 (Histidina 92) para o ligante for maior do que 9,61 Å, a FEB será muito ruim.

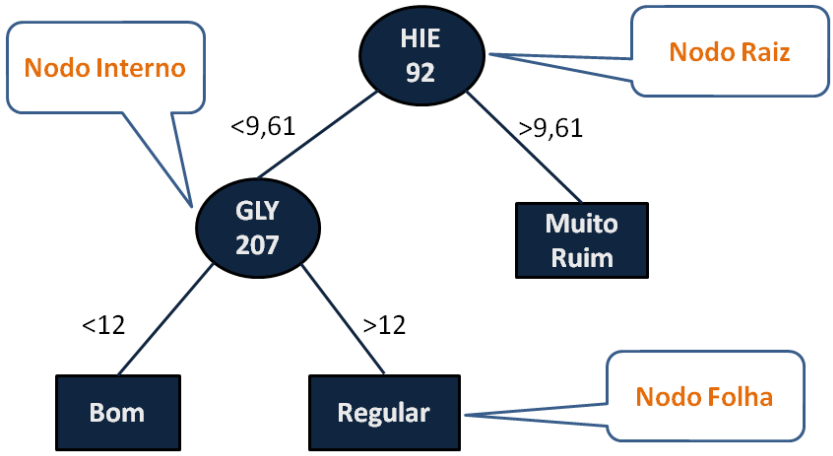


Figura 4.3: Exemplo de árvore de decisão. Os atributos preditivos são distâncias entre os resíduos do receptor e um determinado ligante e o atributo classe é a FEB.

A partir de um mesmo conjunto de atributos há milhares de árvores que podem ser determinadas. Porém, encontrar a árvore que melhor representa a relação entre os atributos tem um alto custo computacional. Dessa forma, têm sido desenvolvidos muitos algoritmos para resolver esse problema. Geralmente esses algoritmos empregam uma estratégia que constrói uma árvore tomando uma série de decisões sobre qual atributo considerar para particionar os dados localmente ótimos. Um desses algoritmos, o Algoritmo de Hunt, é a base de muitos algoritmos de árvores de decisão, incluindo o ID3, C4.5 e o CART.

No algoritmo de Hunt, uma árvore de decisão cresce de uma forma recursiva pelo particionamento de registros de treino em sucessivos subconjuntos. Dado D_t, um conjunto de registros de treino, associados a um nodo t e $y = y_1, y_2, ..., y_c$ os atributos-classes, o algoritmo de Hunt pode ser definido como:

1. se todos os registros em D_t pertencem à mesma classe em y, então t é um nodo folha chamado y_t;

2. se D_t contém registros que pertencem à mais de uma classe, então uma condição de teste é aplicada a um atributo interior selecionado para particionar os registros em subconjuntos menores. Um nodo filho é criado a cada resultado da condição de teste e os registros em D_t são distribuídos nos nodos filhos baseados nos resultados da condição de teste. O algoritmo é então recursivamente aplicado para cada nodo filho.

Muitas considerações devem ser feitas pelo algoritmo de árvore de decisão, que envolve etapas bem mais detalhadas, principalmente, no que diz respeito a escolher o atributo que vai dividir os

dados e, dependendo do tipo de atributo (binário, nominal, ordinal ou contínuo) definem como a classificação deve ser feita.

4.3.1.2 Métricas de Avaliação de Árvores de Decisão

Para avaliar os modelos gerados com árvores de decisão, quando não se tem um conjunto de teste disponível, uma abordagem comum é executar uma validação cruzada (do inglês, *cross-validation*). Na validação cruzada com 10 partições (10-*fold cross-validation*) os dados são divididos randomicamente em 10 partes de tamanhos iguais, onde 9 das 10 partes são utilizadas para aprendizagem e 1 parte para teste. Esse procedimento é repetido 10 vezes, onde a cada execução uma parte diferente do conjunto de entrada é utilizada como conjunto de teste. Então, é calculada a média dos 10 valores das métricas estimados a cada execução para os conjuntos de teste e esses valores servem para estimar a performance dos modelos produzidos [HAL09]. Para árvores de decisão algumas das métricas típicas são, considerando $p_1, p_2, ..., p_n$ como os valores preditos nas instâncias de teste e $a_1, a_2, ..., a_n$ como os valores reais:

- Acurácia (Acc): é a taxa de instâncias que foram classificadas corretamente durante o processo de validação cruzada. Valores maiores de acurácia indicam modelos melhores;
- *Root-Mean Squared Error* (RMSE): Valores menores indicam modelos melhores

$$RMSE = \sqrt{\frac{(p_1 - a_1)^2 + ... + (p_n - a_n)^2}{n}} \tag{4.1}$$

- *Mean Absolute Error* (MAE): calcula a média da magnitude dos erros individuais, sem considerar seus sinais. Valores menores indicam modelos melhores.

$$MAE = \frac{|p_1 - a_1| + ... + |p_n - a_n|}{n} \tag{4.2}$$

- *F-Score* (FM): Considera os valores de precisão e *recall* calculados a partir da matriz de confusão. Na matriz de confusão, determinada com base nos valores de $p_1, p_2, ..., p_n$ e $a_1, a_2, ..., a_n$ tem-se [TAN05]:
  - TP - *True Positive*: corresponde ao número de exemplos verdadeiros positivos corretamente classificados pelo modelo;
  - FN - *False Negative*: corresponde ao número de exemplos positivos erroneamente classificados como negativos pelo modelo;
  - FP - *False Positive*: corresponde ao número de exemplos negativos erroneamente classificados como positivos;
  - TN - *True Negative*: corresponde ao total de exemplos negativos classificados corretamente pelo modelo.

$$Precisao, p = \frac{TP}{TP + FP} \tag{4.3}$$

$$Recall, r = \frac{TP}{TP + FN} \tag{4.4}$$

$$F - score, F = \frac{rp}{\frac{r+p}{2}} \tag{4.5}$$

4.3.2 Regressão

A regressão é uma técnica de modelagem preditiva onde o atributo-classe é contínuo. Modelos de regressão lineares e não-lineares são utilizados em várias áreas como biologia, medicina, agronomia, engenharias. Como exemplos de aplicações para regressão pode-se citar: a previsão de um índice na bolsa de valores, a projeção de vendas de uma empresa, a previsão de quantidade de precipitação, etc.

Segundo Han e Kamber [HAN06] análises de regressão podem ser aplicadas para modelar a relação entre uma ou mais variáveis independentes (variáveis preditoras) e uma variável dependente (que é uma variável contínua, chamada de variável resposta). Este relacionamento pode ser por uma equação linear ou uma função não linear. No contexto de mineração de dados, as variáveis preditoras são os atributos de interesse que descrevem as tuplas. Esses valores, em geral, são conhecidos. A variável resposta é a variável que se deseja predizer o valor. Além do atributo-alvo ser contínuo, a regressão também é uma boa alternativa quando todos os valores dos atributos preditores são contínuos.

A análise de Regressão Linear envolve a variável resposta y e uma simples variável preditora, x. Na forma mais simples de regressão, modela-se y como uma função linear de x:

$$y = b + wx \tag{4.6}$$

onde assume-se como constante a variância de y e b e w são coeficientes de regressão. Esses coeficientes podem ser resolvidos por diferentes métodos que não serão detalhados no contexto deste trabalho. A Regressão múltipla linear é uma extensão da regressão linear, porém envolve mais do que uma variável preditora. Isto permite que a variável y seja modelada como uma função linear de n variáveis preditoras ou atributos. As equações que relacionam y com as n variáveis preditoras se tornam longas e difíceis de serem resolvidas a mão, sendo necessária a aplicação de software para isso.

A Regressão não-linear é aplicada quando não há uma dependência linear entre os dados. Regressão polinomial é geralmente aplicada quando há somente uma variável preditora, o que é modelado adicionando-se termos ao modelo linear. Aplicando-se transformações as variáveis pode-se converter um modelo não-linear em um linear.

Além de regressão linear, regressão linear de múltiplas variáveis e regressão não-linear, há outras formas de regressão que não serão detalhadas neste trabalho, como por exemplo, regressão logística, regressão de Poisson, entre outros. Além disso, há 2 tipos principais de árvores para predição numérica: árvores de regressão e árvores modelo. Na árvore de regressão cada nodo-folha armazena um valor contínuo, que é na verdade uma média dos valores dos atributos preditores das tuplas de teste. Nas árvores modelo, cada nodo-folha contém um modelo de regressão que corresponde a uma equação com múltiplas variáveis para a predição do atributo.

4.3.2.1 Algoritmo M5P - Árvores Modelo

O algoritmo M5P foi desenvolvido por Witten e Frank [WIT05] baseado no algoritmo M5' [WAN97], uma implementação otimizada do clássico algoritmo M5 [QUI92]. Esse algoritmo pode manipular tarefas que envolvem arquivos de entrada de alta dimensionalidade e atributos que podem ser numéricos [QUI92].

O resultado da execução do M5P são as chamadas árvores modelo (Exemplo na Figura 4.4). Esse tipo de árvore é construída pelo algoritmo, inicialmente, como uma árvore de decisão comum. Uma vez que essa árvore básica foi obtida, o algoritmo concentra-se em podar a árvore. A diferença da árvore modelo para outros tipos de árvores de decisão é que os nodos-folha são substituídos por um plano de regressão em vez de um valor constante, como mostra o exemplo de modelo linear a direita na Figura 4.4 que quantifica a contribuição dos atributos preditivos na determinação do atributo-alvo (FEB). No modelo linear, cada parte da equação corresponde a um dos resíduos do receptor (por exemplo, ILE15 é o 15°resíduo do receptor, uma isoleucina) multiplicados por um valor constante que quantifica sua contribuição no valor final do atributo-classe (FEB).

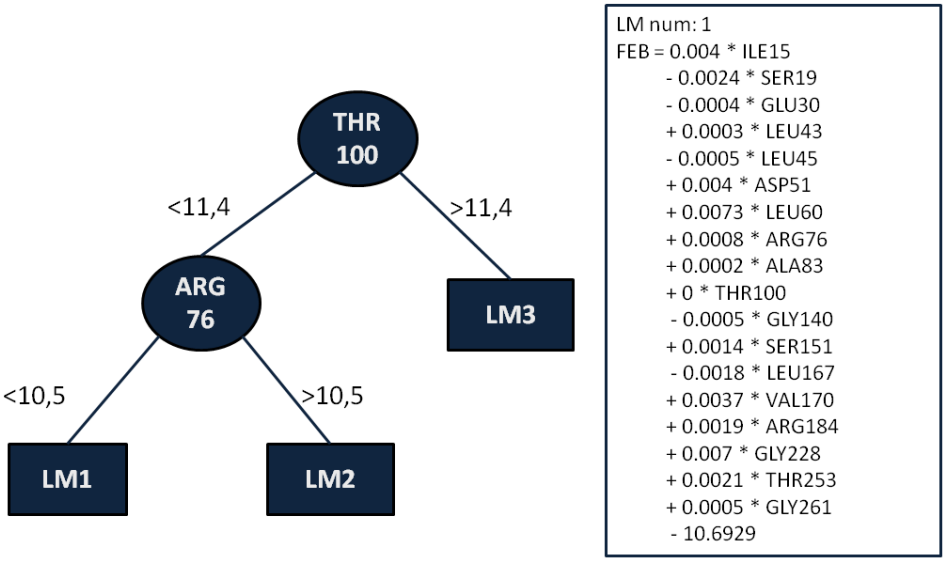


Figura 4.4: Exemplo de árvore modelo. Os atributos preditivos são distâncias entre os resíduos do receptor e o atributo-alvo é a FEB, sendo os modelos lineares definidos em relação a esse valor.

Quando a árvore modelo é utilizada para predizer o valor para uma determinada instância-teste a árvore é percorrida da raiz para as folhas de maneira normal, utilizando os valores dos atributos

das instâncias para tomar decisões de rota em cada nodo. O nodo folha conterá um modelo linear ou *Linear Mode* (LM) que quantifica a contribuição de cada atributo na predição do atributo-classe [WAN97].

4.3.2.2 Métricas de Avaliação de Árvores Modelo

A maioria das métricas de avaliação de árvores modelo são as mesmas das árvores de decisão, como por exemplo as métricas RMSE e MAE (Seção 4.3.1.2). Porém, além das métricas apresentadas anteriormente, é comum para árvores modelo se utilizar a métrica Correlação, que mede a correlação estatística entre a e p, onde valores maiores indicam modelos melhores. Correlação ($Correl$) é definida por:

$$Correl = \frac{S_{pa}}{\sqrt{S_p S_a}} \tag{4.7}$$

onde, S_{pa}, S_p e S_a :

$$S_{pa} = \frac{\sum_i (p_i - \overline{p})(a_i - \overline{a})}{n-1} \tag{4.8}$$

$$S_p = \frac{\sum_i (p_i - \overline{p})^2}{n-1} \tag{4.9}$$

$$S_a = \frac{\sum_i (a_i - \overline{a})^2}{n-1} \tag{4.10}$$

4.3.3 Agrupamento

Segundo Chen *et al.* [CHE96], o processo de agrupar objetos físicos ou abstratos em classes de objetos similares é chamado agrupamento (do inglês, *clustering*). Então os objetos chamados pontos dentro de um grupo (do inglês, *cluster*) tem alta similaridade entre si e alta dissimilaridade de objetos que estejam em outros grupos [HAN06]. Essa medida de similaridade entre objetos é feita baseada nos valores dos atributos que os descrevem.

O agrupamento é uma técnica de aprendizado não supervisionado que permite a identificação de regiões densas e esparsas no espaço de um objeto, descobrindo padrões de distribuição e correlações interessantes entre os dados.

Os grupos apresentam diversas classificações, como por exemplo quanto à maneira como são feitos, que pode ser pela semelhança dos dados ou por dados com mesmo significado (Tan *et al.* [TAN05]). Também apresentam classificações quanto ao tipo de agrupamento, que é o conjunto de todos os grupos (Chen *et al.* [CHE96]). Esses podem ser hierárquicos ou particionados, sobrepostos ou exclusivos, etc.

Há na literatura muitos algoritmos de agrupamento. Apesar das características dos diferentes métodos de agrupamento muitas vezes se sobreporem, é importante que sejam organizados em diferentes categorias. Em geral, segundo Han e Kamber [HAN06], os métodos podem ser classificados em:

(a) Métodos de particionamento: Dada uma base de dados com n objetos, um método de particionamento constrói k partições dos dados, onde cada partição representa um grupo e $k \leq n$. Dado k, o total de partições, esse método utiliza uma técnica de realocação interativa, que busca melhorar o particionamento movendo objetos de um grupo para outro. Os algoritmos de agrupamento mais comuns que utilizam esse método são: ***K-means*** [HAR79], onde cada grupo é representado pelo valor médio dos objetos no grupo; e ***K-medoid***, onde os grupos são representados por um dos objetos localizados próximo do centro do mesmo.

(b) Métodos hierárquicos: Neste método os objetos são decompostos hierarquicamente. O resultado de um algoritmo de agrupamento hierárquico pode ser representado graficamente na forma de uma árvore chamada dendograma. Esta estrutura pode expressar graficamente o processo de união ou divisão entre os grupos e todos seus níveis intermediários. De acordo com a maneira como a decomposição é feita, o método hierárquico pode ser: aglomerativo (*botton-up*), que começa com cada objeto em um grupo para então uni-los até que todos estejam em um grupo único ou que determinada condição seja satisfeita; ou divisivo, (*top-down*) que começa com todos os objetos em um mesmo grupo, e a cada iteração, o grupo é dividido em grupos menores, até que cada objeto esteja em um grupo ou que uma condição de parada seja satisfeita.

(c) Métodos baseados em densidade: A idéia principal desse método de agrupamento é o crescimento de determinado grupo até que a densidade (número de objetos no grupo) na vizinhança exceda um certo limiar, ou seja, para cada objeto de um grupo, a sua vizinhança em um certo raio, tem que conter um número mínimo de objetos.

(d) Métodos baseados em malhas: esse método quantifica o espaço dos objetos em um número finito de células que formam uma estrutura de malha. A maior vantagem dessa abordagem é processamento rápido uma vez que é independente do número de objetos e dependente somente do número de células em cada dimensão.

(e) Métodos baseados em modelo: nesses métodos são definidos modelos hipotéticos para cada um dos grupos e é buscado o melhor encaixe dos dados nos modelos. O algoritmo baseado em modelo então deve estabelecer grupos construindo uma função de densidade que reflita a distribuição dos objetos. Exemplos de algoritmos que implementam esse método são: EM (*expectation-maximization*), COBWEB e SOM (*Self-organized maps*).

A seguir serão descritos os algoritmos de agrupamento utilizados. A descrição dos mesmos é feita principalmente de acordo com suas implementações do trabalho de Shao et al. [SHA07], aplicados durante o desenvolvimento desta Tese (Capítulo 8). Shao et al. classifica os algoritmos de agrupamento implementados em [SHA07] da seguinte forma:

1. Divisivo ou *Top-Down*: Correspondem aos algoritmos hierárquicos divisivos;

2. Aglomerativo ou *Botton-Up:* São os algoritmos hierárquicos aglomerativos; Os diferentes algoritmos aglomerativos diferem entre si na forma de escolha de que pares de grupos serão

combinados a cada execução. Uma vantagem desse método é que as informações sobre como os grupos são combinados a cada execução pode ser salva, possibilitando que uma única execução possa desfazer algum passo já executado;

3. *Refinado*: Nos algoritmos definidos por [SHA07] como refinados, inicia-se por grupos aleatórios e iterativamente os membros de um grupo vão sendo refinados. Nesta classificação estão os algoritmos classificados por [HAN06] como métodos de particionamento e métodos baseados em modelos;

4.3.3.1 Algoritmo *Complete Linkage*

O algoritmo *Complete Linkage*, um algoritmo hierárquico aglomerativo, utiliza a técnica conhecida como Farthest Neighbor ou vizinho mais distante. A seguir é descrito o algoritmo básico de agrupamento hierárquico aglomerativo [TAN05, XU08]:

Algoritmo 4.1: Algoritmo Hierárquico Aglomerativo básico.

1: Inicia n grupos C, cada um com um ponto
2: Calcule uma matriz M de proximidades de acordo com uma função F
3: Repita
4: Localize os grupos C_i e C_j com menor distância em M
5: Construa um novo grupo C_{ij} combinando C_i e C_j
6: Atualiza M para refletir a proximidade entre o novo grupo C_{ij} e os grupos originais
7: Até que todos os pontos estejam em um grupo ou que o número desejado de grupos seja alcançado

Para o algoritmo *Complete Linkage* a distância entre 2 grupos é definida como a maior distância entre um par de pontos individuais, cada um pertencendo a um dos grupos [SHA07]. Dessa forma, a função de distância F entre o grupo combinado C_{ij} e qualquer outro grupo C_l é calculada com a maior distância entre os pontos integrantes de ambos grupos [XU08]:

$$F(C_l,(C_i,C_j)) = \max(F(C_l,C_i),F(C_l,C_j)) \tag{4.11}$$

4.3.3.2 Algoritmo *Edge Linkage* ou *Single Linkage*

É um dos algoritmos hierárquicos aglomerativos mais simples, descritos por Johnson [JOH67]. Esse algoritmo utiliza a técnica do vizinho mais próximo, onde, ao contrário do algoritmo *Complete Linkage*, a distância entre um grupo e outro é definida como a menor distância entre um par de pontos pertencentes a cada um dos grupos. A cada iteração os grupos mais próximos são combinados até que o número de grupos desejado seja atingido [SHA07]. Dessa forma, a função F do algoritmo 4.1 para o *Edge* é [XU08]:

$$F(C_l,(C_i,C_j)) = \min(F(C_l,C_i),F(C_l,C_j)) \tag{4.12}$$

4.3.3.3 Algoritmo *Average Linkage*

As estratégias de distância mínima e máxima, *Edge* e *Complete Linkage* representam dois extremos em termos de distância entre grupos. Como a maioria dos procedimentos que envolvem esses extremos, esses algoritmos tendem a ser altamente sensíveis a *outliers*. Por essa razão foi desenvolvido o algoritmo *Average Linkage*, com o objetivo de ser uma abordagem intermediária. Dessa forma, nesse algoritmo de agrupamento hierárquico aglomerativo a distância utilizada para combinar 2 grupos a cada execução é medida pela média de todas as distâncias calculadas entre pontos individuais dos 2 grupos [SHA07], sendo então F do algoritmo 4.1 definida como [XU08]:

$$F(C_l,(C_i,C_j)) = \frac{1}{2}(F(C_l,C_i) + F(C_l,C_j)) \tag{4.13}$$

4.3.3.4 Algoritmo *Linkage* ou *Centroid Linkage*

O *Linkage* é um algoritmo de agrupamento aglomerativo muito similar ao *Edge Linkage*. Porém no *Linkage*, a distância entre os grupos é definida como a distância entre os centróides dos mesmos [SHA07]. Uma desvantagem desse algoritmo está relacionada ao tamanho dos grupos selecionados para serem combinados a cada execução. Se esse tamanho dos grupos for muito diferente, o centróide do novo grupo será semelhante ao do grupo de maior tamanho, o que ocasionaria a perda de informações do menor grupo.

A função F do algoritmo 4.1 é então definida para o *Linkage* como [XU08], onde n corresponde ao total de pontos pertencentes a determinado grupo:

$$F(C_l,(C_i,C_j)) = \frac{n_i}{n_i+n_j}F(C_l,C_i) + \frac{n_j}{n_i+n_j}F(C_l,C_j) - \frac{n_i n_j}{(n_i+n_j)^2}F(C_i,C_j) \tag{4.14}$$

4.3.3.5 Algoritmo *Centripetal*

Esse é um algoritmo de agrupamento hierárquico aglomerativo derivado do algoritmo CURE [GUH98]. Nesse algoritmo cada cluster é representado por 5 pontos representativos. A escolha dos pontos representativos é feita selecionando os 5 pontos distantes ao máximo dentro do grupo [SHA07]. Em seguida, os pontos representativos são aproximados do centróide do grupo por meio de um fator de encolhimento [GUH98], que neste caso, é realizado movendo-se cada ponto $1/4$ em direção ao centróide do grupo, gerando novos pontos representativos [SHA07]. Esse movimento centrípeto em direção ao centróide tem por objetivo tornar o algoritmo menos sensível a *outliers*. A cada passo de iteração, o par de grupos com os representativos mais próximos são combinados (mesma estratégia utilizada no algoritmo *Edge Linkage*), porém considerando apenas os pontos representativos (e não todos os pontos do grupo) e novos pontos representativos são calculados. A escolha de 5 pontos representativos e do movimento de $1/4$ são escolhas arbitrárias da implementação de Shao et al. [SHA07].

4.3.3.6 Algoritmo *Centripetal Complete*

Esse algoritmo é uma variação do algoritmo Centripetal onde a estrutura do algoritmo é a mesma, porém a cada passo de iteração, o par de grupos com os representativos com maior distância (ou seja, o contrário do algoritmo *Centripetal*) são combinados.

4.3.3.7 Algoritmo *Hierarchical*

É o único algoritmo divisivo utilizado, sendo o de execução mais rápida [SHA07]. Ele inicia sua execução associando todos os pontos a um grande grupo. Então, iterativamente eles dividem esse grande grupo em 2 sub-grupos a cada estágio. Um contador que controla o número de grupos é aumentado de 1 a cada iteração até alcançar o número estabelecido, sendo sensível a *outliers*.

Na implementação desse algoritmo apresentada por Shao et al. [SHA07] foi estabelecido que o diâmetro de um grupo é a distância máxima entre quaisquer 2 pontos neste grupo. Em cada ciclo de execução é encontrado o grupo com maior diâmetro. Este é então dividido em 2 pontos contidos no diâmetro deste grupo. Os pontos são divididos entre esses 2 grupos, de acordo com o diâmetro mais próximo a cada ponto. Agrupamentos hierárquicos podem produzir grupos com tamanhos diferentes, mas não podem produzir grupos com diâmetros muito diferentes.

4.3.3.8 Algoritmo *Bayesian*

Esse algoritmo foi definido por Shao et al. [SHA07] como do tipo *Refinado*. Sua execução inicia com grupos aleatórios (definidos como sementes) com centróides também aleatórios. Os grupos são então refinados utilizando um algoritmo EM (*Expectation-Maximization*) (descrito a seguir por ser a base da implementação do *Bayesian* por Shao et al. [SHA07]). De acordo com os autores uma série de execuções desse algoritmo devem ser realizadas, com diferentes sementes iniciais para que resultados consistentes sejam obtidos.

Algoritmo 4.2: Algoritmo EM utilizado em [SHA07].

1: Define grupos semente e escolhe aleatoriamente centróides para esses grupos
2: Repita
3: Passo Expectation: Para cada ponto é calculada a probabilidade deste pertencer a cada grupo
4: Passo Maximization: Dadas as probabilidades reestima os grupos e centróides de acordo com os pontos reais de forma a maximizar as probabilidades
5: até que os parâmetros não mudem ou que seja atingido um certo limiar

4.3.3.9 Algoritmo *K-means*

O algoritmo *K-means* [HAR79], também classificado por Shao et al. [SHA07] como do tipo *Refinado*, tem como parâmetro de entrada o número de grupos, k, onde um conjunto de n pontos é particionado em k grupos de forma que a similaridade entre os pontos dentro de um mesmo grupo

seja alta e entre grupos diferentes seja baixa. A similaridade é medida de acordo com o valor médio dos pontos no grupo, chamado de centróide ou centro de gravidade. O algoritmo que descreve o *K-means* é o seguinte:

Algoritmo 4.3: Algoritmo *K-means* básico.

1: Selecione K pontos como centróides
2: Repita
3: Forme K grupos atribuindo cada objeto ao centróide mais próximo
4: Calcule o centróide de cada grupos
5: Até que os centroides não se alterem

Para a atribuição de um ponto ao centróide mais próximo, é necessário uma medida de similaridade que quantifique a noção de que centróide é mais perto de cada objeto. Uma das funções de similaridade mais utilizadas é a distância Euclidiana. Geralmente essa medida de similaridade adotada pelo *K-means* é simples, uma vez que o algoritmo calcula repetidamente a proximidade dos pontos para os centróides.

4.3.3.10 Algoritmo *SOM*

Esse algoritmo é classificado por [SHA07] como do tipo *Refinado*. Resumidamente, agrupamento utilizando o algoritmo *SOM* consiste dos passos descritos no Algoritmo 4.4.

Algoritmo 4.4: Algoritmo de agrupamento utilizando *SOM*.

1: Inicializa os centróides aleatoriamente
2: Repita
3: Selecione o próximo ponto
4: Determine o centróide mais próximo desse ponto
5: Atualize esse centróide e todos os outros centróides próximos deste, em uma vizinhança especificada
6: Até que os centroides não se alterem ou que um limiar seja excedido
7: Associe cada ponto ao centróide mais próximo e retorne os centróides dos grupos.

Mapas Auto-organizáveis, do inglês *Self-Organizing Maps* (SOM) é uma técnica para agrupamento e visualização de dados baseada em um ponto de vista de redes neurais [TAN05]. Assim como outros algoritmos de agrupamento baseados em centróide, o objetivo do *SOM* é encontrar um conjunto de centróides e associar cada ponto do conjunto de dados ao centróide que apresentar a melhor aproximação deste ponto. Assim como o K-means, os pontos são processados um por vez e o centróide mais próximo é atualizado [TAN05]. Uma característica diferencial desse algoritmo é que o mesmo impõe uma organização topográfica (espacial) dos centróides. Dessa forma, apesar de semelhante ao K-means, diferencia-se por essa relação topográfica entre os centróides. Durante o processo de treinamento, o *SOM* utiliza cada ponto para atualizar o centróide mais próximo e os demais centróides próximos topograficamente [TAN05].

4.3.3.11 Métricas de Avaliação de Agrupamento

Segundo Shao et al. [SHA07] não há uma métrica de avaliação de agrupamento aceita universalmente. Neste trabalho implementado por [SHA07] são utilizadas principalmente 2 métricas, o *Davies-Bouldin Index* (DBI) e o *pseudo F-statistic* (pSF). O DBI é definido por [DAV79]:

$$DBI = \frac{1}{n} \sum_{i=1, i \neq j}^{n} max(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}) \tag{4.15}$$

onde n é o número de grupos, σ_i é a distância média de todos os pontos do grupo i para seu centróide c_i, σ_j é a distância média de todos os pontos do grupo j para seu centróide c_j e $d(c_i, c_j)$ é a distância entre os centróides c_i e c_j. Menores valores de DBI correspondem a grupos que são compactos e cujos centróides estão distantes uns dos outros.

A métrica pSF é baseada na comparação de variância entre os grupos para a variância residual sobre todos os pontos [SHA07] e é definida por:

$$pSF = \frac{\frac{T - P_G}{G-1}}{\frac{P_G}{n-G}} \tag{4.16}$$

onde G é o número de grupos, n é o número de pontos, T é a soma (para todos os grupos) das distâncias dos pontos para seus centróides e P_G é a distância total de todos os pontos para o centróide de um grupo. Valores mais altos para pSF indicam melhores agrupamentos.

Essas métricas são imperfeitas [SHA07]. Por exemplo, baixos valores de DBI podem ser resultados de agrupamentos que apresentam muitos grupos com somente um ponto. E, pSF tende a resultar em valores altos quando todos os grupos tem aproximadamente o mesmo tamanho, mesmo que estes sejam mal formados. O ideal é utilizar essas métricas em conjunto com uma inspeção visual nos resultados.

4.3.4 Associação

A metodologia de Associação é uma técnica de mineração útil para a descoberta de relações interessantes escondidas em grandes bases de dados [TAN05]. Essas relações podem ser representadas na forma de regras de associação. Por exemplo, utilizando o mesmo contexto dos exemplos de árvore de decisão e árvores modelo acima, e considerando o seguinte conjunto de transações, onde cada linha corresponde aos resíduos do receptor que interagem com determinado ligante, após um experimento de docagem molecular:

1. HIE92, GLY207, THR100
2. ARG76, THR100
3. HIE92, GLY207
4. HIE92, THR100

5. HIE92, GLY207, ARG76

Ao se utilizar Associação pode-se obter regras como:

- HIE92 $\rightarrow$ GLY207

Nesse exemplo, a regra de associação indica que há uma forte relação entre esses 2 atributos (HIE92 e GLY207) e que muitas vezes em que o resíduo Histidina 92 do receptor interage com determinado ligante, o resíduo Glicina 207 também interage. Dessa forma, Tan et al. [TAN05] define regras de associação como uma expressão na forma $X \rightarrow Y$, onde X e Y são conjuntos disjuntos.

4.3.4.1 Métricas de Avaliação

Para avaliar as regras geradas há duas métricas: suporte e confiança. Suporte determina o quão frequente uma regra é aplicável a um conjunto de dados, enquanto que Confiança determina o quão frequente os itens em Y aparecem em transações com X [TAN05]. Ou seja, essas métricas refletem o grau de utilidade e o grau de certeza respectivamente. Tipicamente, regras de associação são consideradas interessantes se elas satisfazem um limiar mínimo de suporte e confiança [HAN06] definidos pelo usuário de acordo com os dados.

4.3.4.2 Algoritmo *Apriori*

O algoritmo *Apriori* foi proposto por Agrawal et al. em [AGR93] para minerar itens frequentes em bases de dados para a determinação de regras de associação entre os itens. Para isso, o *Apriori* executa múltiplas iterações sobre a base de dados de transações. Na primeira iteração é contabilizado o suporte de cada item. Aqueles itens com um suporte individual maior que o suporte mínimo são considerados os itens mais frequentes. Em cada uma das iterações seguintes, sendo k o número da iteração, os itens mais frequentes na iteração $k-1$ são agrupados em conjuntos de k itens, sendo esses considerados itens candidatos. Para os itens candidatos é então contabilizado o seu suporte, e se o mesmo for maior que o suporte mínimo, esses itens candidatos são considerados frequentes. O processo continuará até que o conjunto de itens frequentes seja um conjunto vazio.

4.4 Considerações Finais

Esse capítulo é uma continuação do anterior sobre Materiais e Métodos. Nele foram apresentados conceitos sobre a área de mineração de dados. Inicialmente são descritas duas definições diferentes para mineração de dados, assim como uma ilustração que mostra onde essa etapa está inserida dentro do processo de KDD. Em seguida, são exemplificadas algumas aplicações de mineração de dados na Bioinformática. Na continuação do capítulo são explicadas as técnicas de mineração de dados, e seus respectivos algoritmos, aplicados neste trabalho.

Com os exemplos apresentados durante esse capítulo, juntamente com a seção sobre pré-processamento é possível concluir que esta é uma etapa importante do processo de KDD e que despende muito tempo para ser executada. Além do mais, se os dados não estão organizados, essa etapa pode ser muito mais dispendiosa. Um dos objetivos do desenvolvimento deste trabalho foi, desde o início, um estudo aprofundado da importância da flexibilidade de receptores em docagem molecular, para, após, utilizar esse conhecimento de forma a acelerar esse processo. Por esse motivo decidimos minerar os dados de resultados de docagem molecular com o modelo de receptor FFR. Esses dados, após gerados, estão em diferentes arquivos de saída e sem organização nenhuma. Sendo assim, foi desenvolvido o trabalho descrito em [MAC08b, MAC08a] que foi o ponto de partida de todos os resultados apresentados nos capítulos seguintes desta Tese.

Nesse trabalho de Machado et al. [MAC08b, MAC08a] 40 resultados de docagem com o modelo FFR são processados (receptor InhA e ligante NADH), um número considerado suficiente para testar a metodologia apresentada. Para armazenar esses resultados, assim como as 40 conformações do receptor, foi desenvolvido um banco de dados (BD) descrito na Figura 4.5(a). Nesse BD, todas as informações dos resultados de docagem (FEB, RMSD, etc.) são armazenados na Tabela *Docking*, informações estruturais dos ligantes estão na Tabela *Ligand_atoms* e *Coord_ligand_atoms_docking* e informações estruturais do receptor nas tabelas *Coord_protein_atoms_docking* e *Protein_atoms*.

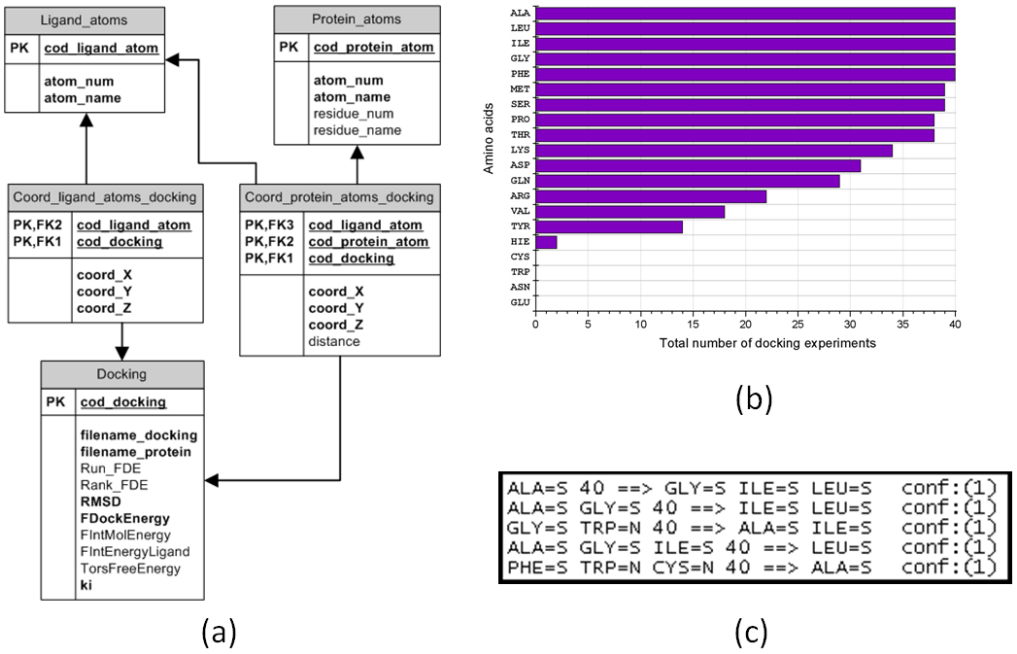


Figura 4.5: Exemplo de Regras de Associação. (a) BD desenvolvido para armazenar 40 resultados de docagem com o modelo FFR. (b) Primeira análise dos dados, a ocorrência de cada tipo de aminoácido a uma distância máxima de 4,0 Å entre receptor-ligante. (c) Algumas regras de associação geradas.

Foram então realizados experimentos com os dados armazenados no BD e a técnica de mineração Associação (algoritmo *Apriori* implementado no WEKA). Para isso foram geradas entradas apropriadas ao WEKA, onde para cada experimento de docagem armazenado é analisada a ocorrência de cada tipo de aminoácido com uma distância máxima de 4,0 Å do ligante. Se alguns dos resíduos

do receptor de determinado tipo (por exemplo, glicina) em algum momento estiver a menos do que 4,0 Å do ligante, é atribuído o valor "Y" para esse tipo de aminoácido, caso contrário é atribuído "N". Assim, nesse arquivo de entrada para o WEKA as instâncias correspondem aos resultados de docagem e os atributos à indicação se determinado tipo de aminoácido interagiu ou não como o ligante naquele experimento. Dessa análise é possível então descobrir quais tipos de aminoácidos mais interagem com o ligante nos 40 resultados analisados. Essa informação então pode ser utilizada para a busca de novos inibidores para esse receptor [MAC08b, MAC08a].

Com esse arquivo preparado, ao abri-lo no WEKA, ainda sem aplicar o *Apriori*, já é possível identificar o número de simulações de docagem que cada um dos tipo de aminoácidos interage com o ligante, conforme mostrado na Figura 4.5(b). A seguir, com o *Apriori*, conseguimos extrair regras como as de exemplo mostradas na Figura 4.5(c). Como esse trabalho inicial foi possível perceber que há muita informação importante sobre a interação receptor-ligante que, sem um processo de KDD, é muito difícil (senão impossível) de extraí-las diretamente dos arquivos de saída da docagem molecular e da DM.

Com a continuação desse trabalho [MAC08b, MAC08a] percebeu-se que esse BD desenvolvido não era o modelo mais apropriado para armazenar grandes quantidades de experimentos de docagem com o FFR, considerando por exemplo todos os *runs* de cada docagem, diferentes ligantes, diferentes FFR, etc. Para isso então é proposto o banco de dados FReDD - *Flexible Receptor Docking Database* para armazenar todos esses resultados de docagem molecular e conformações da trajetória por simulação pela DM que serve para diferentes FFR e diferentes ligantes. O FReDD será descrito no próximo capítulo.

5. RESULTADOS 1 - O BANCO DE DADOS FReDD

Este capítulo apresenta o Banco de dados FReDD (*Flexible Receptor Docking Database*), desenvolvido neste trabalho para armazenar os resultados de docagem molecular com o modelo FFR. A partir dos dados armazenados nesse BD foi possível a utilização de diferentes técnicas de mineração de dados conforme será descrito nos próximos capítulos. Esse BD é uma extensão do modelo apresentado em Machado et al. [MAC08b, MAC08a]. O modelo final do FReDD, seu conteúdo e análises preliminares nos seus dados foram publicados durante o desenvolvimento desta Tese nos seguintes trabalhos:

- como resumo expandido no LNBI-LNCS [WIN09] durante o evento *Brazilian Symposium on Bioinformatics* de 2009;
- como artigo completo na conferência *IADIS International Conference Applied Computing* de 2010 [WIN10a];
- como capítulo do livro *Tópicos em sistemas colaborativos, multimídia, web e banco de dados* de 2010 [WIN10b]. Nesse capítulo de livro são apresentados, de forma bem resumida, vários tópicos presentes nesta Tese, incluindo o BD FReDD. O mesmo foi apresentado como no minicurso intitulado "Processo de KDD aplicado à Bioinformática" durante o Simpósio Brasileiro de Banco de Dados em 2010.

Além do FReDD, esse capítulo também apresenta como o conteúdo armazenado no FReDD foi preparado para ser utilizado com as técnicas de mineração de dados onde é descrito o algoritmo desenvolvido para gerar essas entradas. Ao final desse capítulo, a partir das entradas preparadas para mineração é descrita uma análise preliminar nesses dados, conforme apresentado no artigo [WIN10a].

O modelo do Banco de dados FReDD [WIN09] é mostrado na Figura 5.1 (desenhado com ferramenta *Microsoft Visio*). Atualmente no FReDD estão armazenados todos os resultados dos experimentos de docagem Fase 1 (Seção 3.5.1 do Capítulo 3). Os resultados de docagem Fase 2 somente foram executados ao final do desenvolvimento desta Tese, por esse motivo ainda não tem seus dados no FReDD.

Esse banco de dados é composto atualmente por 17 tabelas contendo um total de 15.814.183 registros. Conforme já mencionado em Materiais e Métodos, o FReDD foi implementado utilizando o SGBD PostgreSQL [STO86] em um ambiente Linux em uma máquina Core 2 Quad com 8 GB de memória RAM. Os dados armazenados em todas tabelas do FReDD foram inseridos através de *scripts Python*. O FReDD [WIN09, WIN10a] foi desenvolvido para armazenar:

- dados referentes a átomos e aminoácidos existentes;
- dados sobre o receptor e ligantes utilizados;

- dados referentes as conformações do receptor utilizado;
- dados referentes aos resultados de docagem molecular utilizando as conformações do receptor armazenadas. É importante ressaltar que os resultados de todas as execuções dentro de cada experimento de docagem molecular foram considerados.

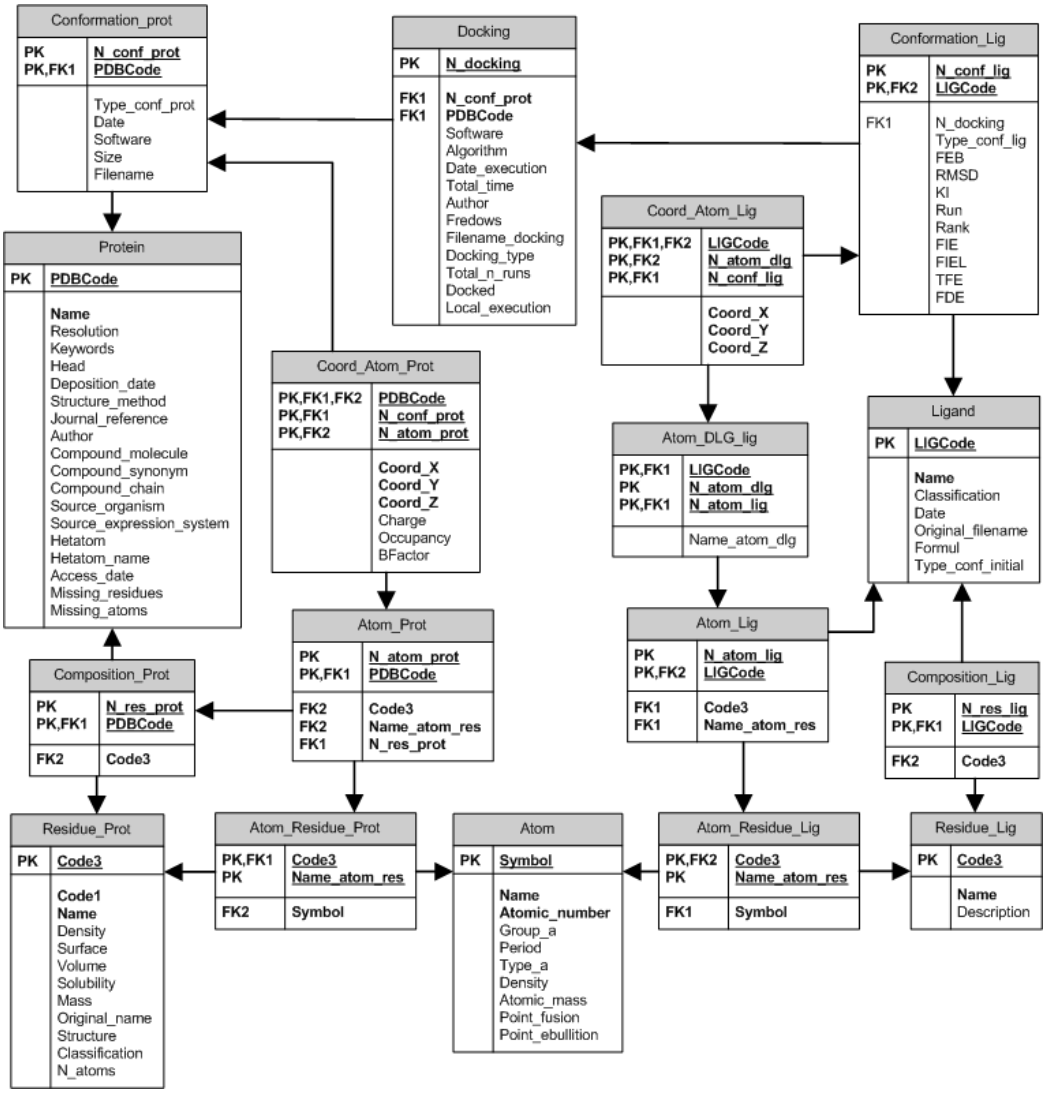


Figura 5.1: Modelo final do banco de dados FReDD.

5.1 Tabelas com Conteúdo Fixo

As tabelas *Atom* e *Residue_Prot* são denominadas de conteúdo fixo. Elas foram definidas dessa forma porque contém informações que tem validade para quaisquer receptores ou ligantes que venham a ser inseridos no FReDD, uma vez que correspondem aos dados referentes aos átomos contidos na tabela periódica e aos aminoácidos naturais e suas variações de nomenclatura de acordo com os software utilizados (como por exemplo o AMBER6.0).

5.1.1 Tabela *Atom*

A tabela *Atom* contém os dados referentes a todos os átomos da tabela periódica, identificados pelo campo *Symbol* - símbolo do átomo. Ela é constituída de 96 registros (os átomos raros não foram incluídos). Essa tabela é importante pois permite que sejam recuperadas informações químicas sobre qualquer átomo que esteja contido no receptor ou em um dos ligantes armazenados, como por exemplo, o grupo e período do átomo na tabela periódica (*Group_a* e *Period*), o nome do mesmo (*Name*) entre outras informações.

5.1.2 Tabela *Residuo_Prot*

Essa tabela contém os dados referentes aos 20 aminoácidos naturais identificados pelo campo *Code3* que corresponde ao código de 3 letras de cada aminoácido. Essa tabela é constituída de 27 registros pois, alguns dos 20 aminoácidos naturais são identificados pelo AMBER6.0 [CAS99] por diferentes códigos de 3 letras. Por exemplo, o aminoácido Histidina pode ser representado pelo código HIS, HIE, HIP e HID de acordo com os diferentes estados de protonação desse resíduo. Assim como a tabela *Atom*, essa tabela é importante caso seja necessário a recuperação de informações químicas sobre determinado aminoácido. Alguns de seus principais campos são:

- *Code1* - o código de uma letra que representa o aminoácido;
- *Name* e *Original_name* - o nome do aminoácido em português e o nome original respectivamente;
- *Classification* - o tipo do aminoácido, como por exemplo, polar, apolar, básico, etc.
- outras informações sobre o resíduo, como por exemplo: *N_atoms*, *Strucuture*, *density*, *surface*, *volume*, *solubility* e *mass*.

5.2 Tabelas com Dados do Receptor

5.2.1 Tabela *Protein*

A tabela *Protein* contém informações sobre os receptores armazenados no banco de dados. A chave primária dessa tabela é o campo *PDBCode* que corresponde ao código PDB do receptor exatamente como é depositado no repositório de estruturas tridimensionais de proteínas PDB [BER00]. Para armazenar esses dados no FReDD foi desenvolvido um *script* em *Python* utilizando uma biblioteca de funções chamada *Biopython* [COC09]. Atualmente essa tabela contém 1 registro, que corresponde aos dados referentes a proteína de estudo nesse trabalho, a InhA (Código PDB:1ENY). Os principais campos contidos nessa tabela são:

- *Name* - nome da proteína;
- *Resolution* - a resolução com que esta estrutura foi determinada;

- *Missing_residues* - se esse campo contém o valor 1, há resíduos de determinada estrutura que não foram identificados experimentalmente, se for 0, a estrutura está completa;
- *Missing_atoms* - se conter 1, há átomos não identificados, se o valor for 0, todos os átomos estão descritos no PDB;
- *Hetatom* - o valor 1 indica que há um ligante nessa estrutura enquanto que *Hetatom* igual a 0 significa que a estrutura contém somente átomos da própria proteína;
- *Hetatom_name* - se há um ligante na estrutura que está sendo armazenada, o nome do mesmo é armazenado nesse campo;
- Informações sobre a obtenção da estrutura: *Deposition_date*, *Structure_method*, *Author*, *Source_organism*, *Source_expression_system*, *Journal_reference* ;
- outras informações contidas no cabeçalho do PDB original do receptor como: *Keywords*, *Head*, *Compound_molecule*, *Compound_synonym* e *Access_date* ;

Parte de um arquivo PDB que descreve uma conformação da proteína está descrito na Figura 5.2. O arquivo PDB original segue o mesmo formato adicionado de um cabeçalho com informações sobre como a estrutura foi obtida, onde está publicada, etc. A primeira linha da Figura 5.2 não existe no arquivo, porém está descrita na Figura 5.2 para auxiliar no entendimento do formato PDB.

	Nro. átomo	Nome átomo	Nome resíduo	Nro. resíduo	Coord. X	Coord. Y	Coord. Z		
ATOM	1	N	ALA	1	15.838	-20.060	8.807	0.00	0.00
ATOM	2	H1	ALA	1	16.368	-19.732	9.602	0.00	0.00
ATOM	3	H2	ALA	1	14.890	-19.724	8.896	0.00	0.00
ATOM	4	H3	ALA	1	15.825	-21.070	8.801	0.00	0.00
ATOM	5	CA	ALA	1	16.474	-19.561	7.583	0.00	0.00
ATOM	6	HA	ALA	1	17.544	-19.764	7.631	0.00	0.00
ATOM	7	CB	ALA	1	15.956	-20.277	6.323	0.00	0.00
ATOM	8	HB1	ALA	1	14.869	-20.209	6.268	0.00	0.00
ATOM	9	HB2	ALA	1	16.404	-19.832	5.435	0.00	0.00
ATOM	10	HB3	ALA	1	16.236	-21.330	6.351	0.00	0.00
ATOM	11	C	ALA	1	16.334	-18.046	7.549	0.00	0.00
ATOM	12	O	ALA	1	16.961	-17.355	8.350	0.00	0.00
...	...	...	...	...	...	...	...	...	...
ATOM	4000	2HD1	LEU	268	-24.705	-18.332	-5.327	0.00	0.00
ATOM	4001	3HD1	LEU	268	-23.418	-18.247	-4.119	0.00	0.00
ATOM	4002	CD2	LEU	268	-25.078	-15.657	-5.622	0.00	0.00
ATOM	4003	1HD2	LEU	268	-24.817	-14.689	-6.051	0.00	0.00
ATOM	4004	2HD2	LEU	268	-25.580	-16.242	-6.392	0.00	0.00
ATOM	4005	3HD2	LEU	268	-25.755	-15.495	-4.783	0.00	0.00
ATOM	4006	C	LEU	268	-21.333	-16.947	-2.983	0.00	0.00
ATOM	4007	O	LEU	268	-21.818	-17.027	-1.828	0.00	0.00
ATOM	4008	OXT	LEU	268	-20.647	-17.857	-3.495	0.00	0.00

Figura 5.2: Parte do arquivo PDB de uma conformação do receptor.

Cada proteína é composta por resíduos, que por sua vez, são compostos por átomos. No caso da InhA, essa proteína é composta por 4.008 átomos (os números e nomes de alguns desses átomos

estão descritos nas colunas 2 e 3 da Figura 5.2 respectivamente) distribuídos em 268 resíduos (o nome e o número de alguns resíduos estão descritos nas colunas 4 e 5 da Figura 5.2). O mesmo átomo pode aparecer mais de uma vez dentro de uma proteína e até mesmo, mais de uma vez dentro do mesmo resíduo, por isso, é sempre identificado por um nome mais um número. O mesmo acontece com os resíduos. O mesmo resíduo pode aparecer inúmeras vezes dentro de uma proteína portanto identificado pelo seu código de 3 letras e por um número. Os dados sobre os átomos e resíduos de cada conformação da proteína são armazenados nas tabelas descritas a seguir.

5.2.2 Tabela *Conformation_Prot*

A tabela *Conformation_Prot* armazena informações referentes às conformações do receptor. Essa tabela é identificada pelos campos *PDBCode* e *N_Conf_Prot* que correspondem ao código PDB do receptor e ao número da conformação. Essa tabela é composta pelos seguintes campos que armazenam:

- *PDBCode* e *N_Conf_Prot*
- *Type_conf_prot* - Tipo de conformação: se é cristalográfica ou resultante de uma simulação pela DM;
- *Filename* - Nome do arquivo;
- *Date*, *Software*, *Size* - dados referentes a conformações resultantes de uma simulação pela DM, *Date* armazena a data que a simulação foi executada, *Software* indica que software foi utilizado para a execução da simulação e *Size* armazena o tamanho da mesma.

Essa tabela contém atualmente 3.100 registros, que correspondem às 3.100 conformações do receptor InhA (*PDBCode* = 1ENY) utilizadas nesse trabalho resultantes da simulação pela DM descrita no Capítulo 2.

5.2.3 Tabela *Composition_Prot*

Essa tabela contém uma lista de todos os resíduos contidos em cada proteína e os relaciona com os aminoácidos naturais armazenados na tabela *Residue_Prot*. Atualmente contém 268 registros que correspondem aos 268 resíduos que a única proteína armazenada contém (1ENY). Esse total de resíduos pode ser visto na Figura 5.2 na coluna 5, onde o PDB inicia pelo Resíduo 1, uma Alanina (ALA) e termina pelo resíduo Leucina (LEU), que corresponde ao resíduo 268. Os principais campos dessa tabela são: *N_res_prot* (número do resíduo na proteína), *PDBCode* (o código identificador da proteína) e *Code3* (o código de 3 letras de cada resíduo).

5.2.4 Tabelas *Atom_Prot* e *Atom_Residue_Prot*

Essas duas tabelas servem para relacionar a proteína que está sendo armazenada com os átomos contidos na mesma, assim como relacionar os átomos com os resíduos.

A tabela *Atom_Prot* armazena as informações contidas nas colunas 2, 4 e 5 do arquivo PDB descrito na Figura 5.2 (Número do átomo, Código de 3 letras do resíduo e Número do Resíduo, respectivamente) e as relaciona com o código PDB da proteína. Como cada átomo aparece somente uma vez dentro de cada proteína, cada registro nessa tabela é identificado pelo código PDB da proteína e pelo número do átomo na mesma (*PDBCode, N_atom_prot*). Essa tabela é necessária porque muitas vezes o mesmo átomo aparece inúmeras vezes dentro de um mesmo PDB da proteína e a cada vez que ele aparece corresponde a um diferente átomo (por exemplo, o 'O' do PDB descrito em parte na Figura 5.2 na primeira vez ele é o átomo 12 do resíduo 1 e a segunda vez é o átomo 4007 do resíduo 268). Então, essa tabela faz esse tipo de relação, do número do átomo com o número do resíduo, para uma determinada proteína. Ela contém atualmente 4.008 registros, que correspondem aos 4.008 átomos da proteína InhA (Código PDB: 1ENY), a única proteína armazenada até o momento.

A tabela *Atom_Residue_Prot* relaciona os átomos contidos em cada tipo de resíduo. Por exemplo, entre as Alaninas contidas nos PDBs que utilizamos nesse trabalho, o número máximo de átomos que cada uma continha era 12 ou menos, os que estão listados na Figura 5.2 das linhas 1 a 12. A diferença no número de átomos de um resíduo dentro de um mesmo arquivo PDB ocorre porque alguns átomos do resíduo podem estar ligados a outros átomos dependendo da posição em que se encontram dentro da estrutura no momento que a simulação pela DM. Portanto a tabela relaciona cada resíduo com o nome e número de átomos que o mesmo pode conter. A tabela contém atualmente 345 registros pois cada resíduo é cadastrado somente uma vez.

Para inserir dados nas tabelas *Atom_Prot* e *Atom_Residue_Prot* foram desenvolvidos *scripts* em *Python*. Esses *scripts* são executados somente 1 vez para cada receptor, uma vez que independente da conformação, um receptor conterá sempre o mesmo número de átomos e de resíduos e a relação que há entre átomos e resíduos não se altera da mesma forma.

5.2.5 Tabela *Coord_Atom_Prot*

Essa tabela é utilizada para armazenar as coordenadas de todos os átomos de cada conformação da proteína. Seus principais campos são:

- *PDBCode*, *N_Conf_Prot* e *N_Atom_Prot* que armazenam o código PDB da proteína, o número de sua conformação e o número do átomo, respectivamente. Esses campos são utilizados em conjunto como chave primária e permitem o relacionamento desta tabela com as tabelas *Conformation_Prot* e *Atom_Prot*;

- *Coord_X*, *Coord_Y* e *Coord_Z* armazenam as coordenadas x, y e z de cada átomo. Um exemplo pode ser visto nas colunas 6, 7 e 8 da Figura 5.2;

- *Charge1*, *Charge2*, *Occupancy* e *BFactor* são campos utilizados no armazenamento de estruturas cristalográficas porque não constam em PDBs resultantes de uma simulação pela DM.

Atualmente somente as conformações resultantes da DM é que estão sendo armazenadas no FReDD.

Essa tabela é composta atualmente 12.424.800 de registros, que correspondem às 3.100 conformações da proteína InhA multiplicadas pelos 4.008 átomos contidos em cada conformação. Para a inserção de dados nessa tabela foi desenvolvido um *script* em *Python* para inserir os dados referentes a cada conformação, primeiro armazenando os dados sobre a conformação na tabela *Conformation_Prot* e após sobre as coordenadas de cada átomo da mesma na tabela *Coord_Atom_Prot*.

5.3 Tabelas com Dados dos Ligantes e de Docagem Molecular

Na Figura 5.1 que descreve o Modelo final do banco de dados FReDD a tabela *Docking* está relacionada às simulações de docagem molecular enquanto que as tabelas *Tabelas Ligand*, *Composition_Lig*, *Residue_Lig*, *Atom_Res_Lig*, *Atom_Lig*, *Atom_DLG_Lig*, *Conformation_Lig* e *Coord_Atom_Lig* estão relacionadas com dados sobre os ligantes e suas conformações.

5.3.1 Tabela *Ligand*

A Tabela *Ligand* é utilizada para armazenar informações referentes aos ligantes. Cada registro é identificado pelo campo *LIGCode*, que corresponde a um código de até 10 letras que identifica o ligante, semelhante ao código PDB que identifica a proteína. Porém, no caso do ligante, esse código não permite uma busca direta do ligante em um banco de dados de pequenas moléculas pois, muitos ligantes são identificados com um mesmo código e uma busca pelo ligante deve ser feita utilizando uma combinação de parâmetros. Além do *LIGCode*, essa tabela contém os campos

- *Name* - Nome do ligante;
- *Classification* - os ligantes podem ser classificados de acordo com suas características eletrônicas e da interação;
- *Date*, *Original_filename* e *Type_conf_initial* - informações sobre o arquivo original do ligante, a data de acesso, o nome do arquivo orginal e o tipo do arquivo original (pode ser por exemplo PDB ou MOL2;
- *Formul* - A fórmula química do ligante.

A Tabela *Ligand* é uma versão inicial já que o foco deste trabalho não é seleção de ligantes e atualmente estamos trabalhando com um número reduzido (somente 4 ligantes). Porém, o laboratório LABIO tem como um de seus objetivos trabalhar com *Virtual Screening* e para isso está sendo estudado por outro membro do laboratório técnicas de seleção de ligantes, trabalho que possivelmente atualizará essa tabela tornando a mesma mais completa e apropriada.

Atualmente a tabela *Ligand* contém atualmente 4 registros, um para cada ligante utilizado nesse trabalho: NADH, PIF, TCL e ETH (descritos no Capítulo 2).

5.3.2 Tabelas *Composition_Lig* e *Residue_Lig*

O banco de dados foi modelado de forma a ser o mais flexível possível, sendo assim, poderá armazenar ligantes compostos por um ou mais resíduos. Essa informação é armazenada nas tabelas *Composition_Lig* e *Residue_Lig*.

A tabela *Composition_Lig* é composta pelos campos *LIGCode* (código identificador do ligante), *N_res_lig* (número do resíduo no ligante) e *Code3* (código identificador do resíduo do ligante) e contém uma lista dos resíduos que compõem cada ligante. A tabela *Residue_Lig* armazena informações sobre os resíduos de cada ligante e contém os campos *Code3* (código identificador do resíduo do ligante), *Name* e *Description* (armazenam informações mais detalhadas sobre os resíduos que compõem os ligantes).

Contudo, atualmente os 4 ligantes de trabalho do LABIO que estão armazenados no FReDD são compostos somente por um resíduo cada. Sendo assim, as tabelas *Composition_Lig* e *Residue_Lig* são compostas atualmente somente por 4 registros cada uma e para facilitar, esses resíduos receberam como identificador *Code3* o próprio *LIGCode* de cada ligante.

5.3.3 Tabelas *Atom_Res_Lig*, *Atom_Lig* e *Atom_DLG_Lig*

As tabelas *Atom_Res_Lig*, *Atom_Lig* e *Atom_DLG_Lig* armazenam informações sobre os átomos que compõem cada ligante, suas relações com os resíduos (ou resíduo) e a relação que há entre os átomos do ligante antes (arquivo MOL2 do ligante) e após a preparação do mesmo para docagem molecular.

A tabela *Atom_Lig* tem como chave primária o número de cada átomo do ligante (campo *N_atom_lig*) e o *LIGCode*. Também armazena o nome do átomo (*Name_atom_res*) e o código de três letras do resíduo de cada átomo (*Code3*). Essa tabela contém atualmente 144 registros, 71 estão relacionados aos átomos que compõem o NADH, 24 registros correspondem aos 24 átomos do ligante TCL em sua versão original (MOL2), 28 registros estão relacionados aos 28 átomos da versão original do ligante PIF e 21 aos átomos do ETH.

Como esses ligantes contém somente 1 resíduo cada, a tabela *Atom_Res_Lig* que relaciona os átomos aos tipos de átomos (Tabela *Atom*) e aos resíduos (através dos campos *Symbol*, *Code3* e *Name_atom_res*) contém o mesmo número de registros da tabela que contém somente os átomos, totalizando então os mesmos 144 registros.

O arquivo original do ligante TCL está descrito na Figura 5.3 como um exemplo de arquivo MOL2 de um ligante, nesse caso, obtido a partir do banco de dados de pequenas moléculas ZINC [IRW05]. Atualmente, a maioria dos ligantes já podem ser obtidos no formato MOL2, o formato ideal, pronto para ser utilizado para execução de docagem molecular. Esse formato consiste no formato PDB mais uma coluna que contém a carga de cada átomo do ligante (corresponde a penúltima coluna da Figura 5.4), coluna essencial para a execução de docagem molecular.

Antes da execução das simulações de docagem molecular é necessário preparar os arquivos do ligante. Para isso, é executado pelo programa *deftors* do AutoDock3.0.5. O arquivo PDBQ do

ligante, preparado para a docagem molecular, está descrito na Figura 5.4.

```
@<TRIPOS>MOLECULE
TCL_DOCKED201
   24    25     1     0     1
SMALL
USER_CHARGES
@<TRIPOS>ATOM
      1 OO1O       -6.0050    0.3620   -2.7050 O.3       1 TCL270     -0.5980 ****
      2 HO1O       -5.0851    0.2406   -2.2578 H         1 TCL270      0.4610 ****
      3 CA1        -6.9940   -0.0100   -1.8250 C.2       1 TCL270      0.2890 ****
      4 CA2        -7.8950   -1.0570   -2.1260 C.2       1 TCL270     -0.3140 ****
      5 HA2        -7.8059   -1.6128   -3.0594 H         1 TCL270      0.1850 ****
      6 CA3        -8.9110   -1.3770   -1.2150 C.2       1 TCL270      0.0300 ****
      7 CLL3      -10.0356   -2.6986   -1.5637 Cl        1 TCL270     -0.1010 ****
      8 CA4        -9.0340   -0.6610   -0.0230 C.2       1 TCL270     -0.1050 ****
      9 HA4        -9.8306   -0.9060    0.6795 H         1 TCL270      0.1540 ****
     10 CA5        -8.1270    0.3780    0.2770 C.2       1 TCL270     -0.3200 ****
     11 HA5        -8.2121    0.9359    1.2095 H         1 TCL270      0.2090 ****
     12 CA6        -7.1150    0.6910   -0.6330 C.2       1 TCL270      0.3510 ****
     13 O7         -6.1830    1.6390   -0.3400 O.3       1 TCL270     -0.4370 ****
     14 CA8        -6.3920    2.9250   -0.7440 C.2       1 TCL270      0.6300 ****
     15 CA9        -5.4680    3.9230   -0.3160 C.2       1 TCL270     -0.2370 ****
     16 CLL9       -4.0944    3.5047    0.7190 Cl        1 TCL270     -0.0660 ****
     17 CA10       -5.6680    5.2590   -0.6980 C.2       1 TCL270      0.0940 ****
     18 HA10       -4.9685    6.0323   -0.3806 H         1 TCL270      0.1290 ****
     19 CA11       -6.7910    5.6030   -1.4990 C.2       1 TCL270     -0.1810 ****
     20 CL11       -7.0544    7.2933   -1.9531 Cl        1 TCL270     -0.0960 ****
     21 CA12       -7.7090    4.6090   -1.9230 C.2       1 TCL270      0.1290 ****
     22 HA12       -8.5658    4.8841   -2.5380 H         1 TCL270      0.1230 ****
     23 CA13       -7.5080    3.2680   -1.5440 C.2       1 TCL270     -0.5460 ****
     24 HA13       -8.2068    2.4955   -1.8651 H         1 TCL270      0.2150 ****
```

Figura 5.3: Parte do arquivo MOL2 do ligante TCL.

```
REMARK  0 active torsions (identified by new id numbers):
REMARK  status: ('A' for Active; 'I' for Inactive)
REMARK       I      3-  1 between atoms:  CA1  and  OO1O
REMARK       I      9- 10 between atoms:  CA6  and  O7
REMARK       I     10- 11 between atoms:  O7   and  CA8
ROOT
ATOM      1  OO1OTCL     1      -6.005   0.362  -2.705  0.00  0.00    -0.598
ATOM      2  HO1OTCL     1      -5.085   0.241  -2.258  0.00  0.00     0.461
ATOM      3  AA1 TCL     1      -6.994  -0.010  -1.825  0.00  0.00     0.289
ATOM      4  AA2 TCL     1      -7.895  -1.057  -2.126  0.00  0.00    -0.129
ATOM      5  AA3 TCL     1      -8.911  -1.377  -1.215  0.00  0.00     0.030
ATOM      6  c3  TCL     1     -10.036  -2.699  -1.564  0.00  0.00    -0.101
ATOM      7  AA4 TCL     1      -9.034  -0.661  -0.023  0.00  0.00     0.049
ATOM      8  AA5 TCL     1      -8.127   0.378   0.277  0.00  0.00    -0.111
ATOM      9  AA6 TCL     1      -7.115   0.691  -0.633  0.00  0.00     0.351
ATOM     10  O7  TCL     1      -6.183   1.639  -0.340  0.00  0.00    -0.437
ATOM     11  AA8 TCL     1      -6.392   2.925  -0.744  0.00  0.00     0.630
ATOM     12  AA9 TCL     1      -5.468   3.923  -0.316  0.00  0.00    -0.237
ATOM     13  c9  TCL     1      -4.094   3.505   0.719  0.00  0.00    -0.066
ATOM     14  AA10TCL     1      -5.668   5.259  -0.698  0.00  0.00     0.223
ATOM     15  AA11TCL     1      -6.791   5.603  -1.499  0.00  0.00    -0.181
ATOM     16  c11 TCL     1      -7.054   7.293  -1.953  0.00  0.00    -0.096
ATOM     17  AA12TCL     1      -7.709   4.609  -1.923  0.00  0.00     0.252
ATOM     18  AA13TCL     1      -7.508   3.268  -1.544  0.00  0.00    -0.331
ENDROOT
TDOF 3
```

Figura 5.4: Arquivo PDBQ do ligante TCL preparado para docagem.

Como é possível analisar, comparando o arquivo do ligante antes da preparação para a docagem e após (Figuras 5.3 e 5.4), há uma diferença em relação ao número de átomos e ao nome de cada átomo. De um total de 24 átomos antes da preparação para a docagem molecular (Figura

5.3) tem-se 18 átomos no arquivo do ligante (Figura 5.4). Como ambas conformações de cada ligante constam no banco de dados, é necessário que a relação que há entre os átomos antes e após a preparação da docagem molecular seja também armazenada. Para isso, desenvolveu-se a Tabela *Atom_DLG_Lig*, composta pelo número de cada átomo do ligante na versão antes da preparação para a docagem molecular (N_atom_lig) e o correspondente número e nome de cada átomo (*N_atom_dlg* e *Name_atom_dlg*) após a preparação do mesmo (essa numeração dos átomos do PDBQ é a mesma que aparece nos arquivos de saída do AutoDock3.0.5). Assim como as tabelas *Atom_Res_Lig* e *Atom_Lig*, a tabela *Atom_DLG_Lig* contém 144 registros, 71 relacionados ao NADH, 24 ao TCL, 28 ao PIF, e 21 à ETH.

A Tabela 5.1 mostra uma parte do conteúdo da tabela *Atom_DLG_Lig* para exemplificar como ocorre a relação entre o arquivo do ligante antes e após a preparação para a docagem. Na primeira coluna da tabela está o código do ligante (*LIGCode*), ou seja, sua identificação dentro da tabela *Atom_Lig*. Na segunda coluna está o campo *Code3*, campo identificador das tabelas *Atom_Res_Lig*, *Atom_Lig* e *Atom_DLG_Lig*. A terceira e quarta colunas listam os campos *N_atom_lig* e *Name_atom_res* que correspondem ao número e nome do átomo do ligante em sua versão original, respectivamente. A quinta coluna mostra o campo *N_atom_dlg*, que se refere ao número do átomo no arquivo DLG.

Tabela 5.1: Parte do conteúdo da tabela *Atom_DLG_Lig* que relaciona o nome e número do átomos nos arquivos do ligante antes e após a preparação para a docagem molecular.

LIGCode	*Code3*	*N_atom_lig*	*Name_atom_res*	*N_atom_dlg*
TCL	TCL	1	OO10	1
TCL	TCL	2	HO10	2
TCL	TCL	...	...	...
TCL	TCL	23	CA13	18
TCL	TCL	24	HA13	18

Como pode-se verificar com os dados descritos na Tabela 5.1, alguns átomos são os mesmos em ambos arquivos do ligante, por exemplo os átomos 1 e 2 são iguais no arquivo original do ligante (colunas 3 e 4 na Tabela 5.1) e no arquivo após a preparação para a docagem (Coluna 5 na Tabela 5.1). Porém, os átomos 23 e 24 do arquivo do ligante TCL antes da preparação para a docagem molecular foram compactados em um átomo no arquivo do ligante final, o átomo 18. Por isso, na tabela *Atom_DLG_Lig* esse átomo do ligante final se relaciona com 2 átomos do arquivo do ligante original.

Os dados das tabelas *Ligand*, *Composition_Lig* e *Residue_Lig* foram inseridos utilizando *scripts Python*. Esses *scripts* foram executados somente 1 vez para cada ligante armazenado.

5.3.4 Tabela *Docking*

A tabela *Docking* armazena informações sobre os experimentos de docagem molecular executados, atribuindo um número sequencial a cada experimento inserido no banco de dados e relacionando

esse número com o número da conformação do receptor utilizado. Os principais campos dessa tabela são:

- *N_Docking* - é um número sequencial utilizado como campo identificador desta tabela;
- *PDBCode* e *N_Conf_Prot* - são os campos que estabelecem o relacionamento da tabela *Docking* com a tabela de conformações do receptor, indicando que conformação do receptor foi utilizado em cada uma das simulações de docagem armazenadas;
- *Software*, *Algorithm*, *Data_Execution*, *Total_Time*, *Author*, *Local_execution*, *Docked* e *Filename_docking* - informações técnicas sobre cada experimento de docagem importantes para registro sobre os experimentos que poderá ser útil em futuras comparações entre resultados;
- *FReDoWs* - esse campo indica se o experimento de docagem foi executado utilizando o *workflow* científico descrito em [MAC07] ou foi executado com o auxílio de *scripts*;
- *Docking_Type* - armazena o tipo de experimento de docagem executado, se foi exaustivo, considerando todos as conformações do receptor ou seletivo, onde somente uma parte das conformações é considerada;
- *Total_N_runs* - armazena o total de *runs* de cada experimento;

Essa tabela contém 12.400 registros, 3.100 registros para cada um dos ligantes NADH, PIF, TCL e ETH. Os *scripts Python* desenvolvidos para armazenar os dados da tabela *Docking* foram executados e os dados das tabelas *Conformation_Lig* e *Coord_Atom_Lig* foram inseridos durante a execução deste mesmo *script*, uma vez que eles se referem aos dados dos mesmos arquivos de saída (os arquivos DLG resultantes da docagem).

5.3.5 Tabelas *Conformation_Lig* e *Coord_Atom_Lig*

As tabelas *Conformation_Lig* e *Coord_Atom_Lig* armazenam os dados referentes aos resultados de cada experimento de docagem molecular.

Para um melhor entendimento do conteúdo dessas tabelas, uma parte do arquivo de saída do software AutoDock3.0.5 que corresponde a 1 *run* executado, está descrito na Figura 5.5 . Esse arquivo é resultante da execução do experimento de docagem molecular considerando-se a conformação 2 do receptor InhA e o ligante TCL.

É na tabela *Conformation_Lig* que estão armazenados os dados marcados em rosa na Figura 5.5, com exceção dos dados sobre as coordenadas do ligante em cada *run* executado. Os principais campos dessa tabela são:

- *LIGCode* e *N_Conf_Lig* - campos utilizados como identificadores. Se *N_Conf_Lig* for igual a 0 (zero) indica que determinado registro de conformação do ligante corresponde a original (MOL2);

- *N_docking* - relaciona a conformação do ligante que está sendo armazenada com o experimento de docagem onde a mesma foi obtida. Se for a conformação inicial de um determinado ligante, o valor de *N_docking* é vazio;
- *Type_conf_lig* - indica o tipo de conformação do ligante, se for 0, trata-se de uma conformação inicial, se for 1, uma conformação resultante de docagem molecular;
- *Run* e *Rank* de cada *run* executado dentro de cada experimento de docagem molecular;
- *RMSD* (*Root Mean Square Deviation*) - corresponde à distância do centro de massa do ligante na sua posição inicial, antes da execução da docagem molecular e o centro de massa do ligante após a execução da docagem;
- *FEB* - Energia de interação final;
- *Ki* - Constante de inibição;
- *FIE* (*Final Intermolecular Energy*), *FIEL* (*Final Internal Energy of Ligand*), *TFE* (*Torsional Free Energy*) e *FDE* (*Final Docked Energy*) - energias envolvidas na ligação;

```
USER    Run = 9
USER    Cluster Rank = 1
USER    Number of conformations in this cluster = 7
USER
USER    RMSD from reference structure       = 8.263 A
USER
USER    Estimated Free Energy of Binding    =   -8.22 kcal/mol  [=(1)+(3)]
USER    Estimated Inhibition Constant, Ki   =   +9.36e-07       [Temperature = 298.15 K]
USER
USER    Final Docked Energy                 =   -8.22 kcal/mol  [=(1)+(2)]
USER
USER    (1) Final Intermolecular Energy     =   -8.22 kcal/mol
USER    (2) Final Internal Energy of Ligand =   +0.00e+00 kcal/mol
USER    (3) Torsional Free Energy           =   +0.00e+00 kcal/mol
USER
USER
USER    DPF = InputFile_SA.dpf
USER    NEWDPF move  LIGmoved.pdbq
USER    NEWDPF about -6.183000 1.639000 -0.340000
USER    NEWDPF tran0 -1.980195 8.080871 4.610087
USER    NEWDPF quat0 0.120343 0.581732 0.804429 -78.162220
USER
USER                  Rank         x       y       z       vdW    Elec       q     RMS
ATOM      1  0010TCL     1      -0.434   7.404   6.709    -0.06  -0.06  -0.598   8.263
ATOM      2  H010TCL     1      -1.146   8.043   7.088    +0.08  +0.05  +0.461   8.263
ATOM      3  AA1 TCL     1      -1.050   6.405   5.992    -0.49  +0.02  +0.289   8.263
ATOM      4  AA2 TCL     1      -0.908   5.044   6.348    -0.54  -0.00  -0.129   8.263
ATOM      5  AA3 TCL     1      -1.525   4.058   5.566    -0.53  -0.00  +0.030   8.263
ATOM      6  c3  TCL     1      -1.376   2.350   6.007    -0.62  -0.02  -0.101   8.263
ATOM      7  AA4 TCL     1      -2.267   4.419   4.440    -0.48  -0.00  +0.049   8.263
ATOM      8  AA5 TCL     1      -2.413   5.780   4.093    -0.30  +0.01  -0.111   8.263
ATOM      9  AA6 TCL     1      -1.799   6.758   4.878    -0.41  -0.01  +0.351   8.263
ATOM     10  07  TCL     1      -1.980   8.081   4.610    +0.04  +0.06  -0.437   8.263
ATOM     11  AA8 TCL     1      -1.082   8.728   3.812    -0.49  -0.01  +0.630   8.263
ATOM     12  AA9 TCL     1      -1.361  10.076   3.442    -0.54  +0.00  -0.237   8.263
ATOM     13  c9  TCL     1      -2.836  10.883   3.994    -0.50  -0.00  -0.066   8.263
ATOM     14  AA10TCL     1      -0.467  10.761   2.605    -0.58  +0.00  +0.223   8.263
ATOM     15  AA11TCL     1       0.701  10.103   2.130    -0.67  -0.01  -0.181   8.263
ATOM     16  c11 TCL     1       1.810  10.962   1.052    -0.84  -0.01  -0.096   8.263
ATOM     17  AA12TCL     1       0.976   8.762   2.499    -0.66  +0.05  +0.252   8.263
ATOM     18  AA13TCL     1       0.082   8.074   3.342    -0.61  -0.07  -0.331   8.263
```

Figura 5.5: Parte do arquivo de saída do programa Autodock. Essa parte compreende o resultado de uma execução (*run*) para o ligante TCL considerando a conformação 2 do receptor.

É necessário que a cada conformação do ligante armazenada na tabela *Conformation_Lig* estejam relacionadas as coordenadas de todos seus átomos. Para isso foi desenvolvida a tabela *Coord_Atom_Lig* cujos principais campos são:

- *LIGCode*, *N_conf_lig* e *N_atom_dlg* - são os campos que formam a chave primária de cada registro. Caso a conformação que esteja sendo armazenada seja a conformação inicial do ligante, o *N_atom_dlg* terá o mesmo valor do *N_atom_lig* da tabela *Atom_DLG_lig*;

- *Coord_X*, *Coord_Y* e *Coord_Z* - que correspondem as coordenadas x, y e z de cada átomo;

A Tabela 5.2 resume o conteúdo de *Conformation_Lig* e *Coord_Atom_Lig*. Na primeira coluna tem-se o nome dos ligantes, a segunda coluna descreve o total de átomos originais de cada ligante enquanto que a terceira coluna contém o total de átomos de cada ligante após a preparação para a docagem molecular. A quarta coluna apresenta o total de registros da Tabela *Conformation_Lig* e a última coluna o total de registros da Tabela *Coord_Atom_Lig*.

Para cada ligante o total de registros na Tabela *Conformation_Lig* é calculado da forma: (resultados de docagem molecular que convergiram para valores válidos para esse ligante) * 10 (Total de *runs* executados em cada uma das simulações de docagem molecular) + 1 conformação inicial do ligante (corresponde a conformação com *N_Conf_Lig*=0).

O total de registros de cada ligante na Tabela *Coord_Atom_Lig* é calculado: [(1 conformação inicial) * (total de átomos do ligante antes da preparação para a docagem)] + [(conformações resultantes da docagem) * (total de átomos desse ligante após a preparação para a docagem)]

Tabela 5.2: Resumo dos conteúdos das Tabelas *Conformation_Lig* e *Coord_Atom_Lig*.

Ligantes (Ligand)	Átomos (Atom_Lig)	Átomos docagem	Conformações (Conformation_Lig)	Coordenadas (Coord_Atom_Lig)
NADH	71	52	3.100*10 + 1 = 31.001	1*71 + 31.000*52 = 1.612.071
PIF	28	24	3.042*10 + 1 = 30.421	1*28 + 30.420*24 = 730.108
TCL	24	18	2.837*10 + 1 = 28.371	1*24 + 28.370*18 = 510.684
ETH	21	13	3.043*10 + 1 = 30.431	1*21 + 30.430*13 = 395.611
Total			120.224	3.248.474

5.4 Etapa de Preparação dos Dados

Após a descrição de todo o modelo do FReDD assim como todo o seu conteúdo, é necessário descrever como esses dados foram pré-processados para os experimentos de mineração de dados. Todos os passos envolvidos nesse pré-processamento são ilustrados na Figura 5.6:

5.4.1 (1) Determinação de Cada Atributo do Arquivo de Entrada

Um dos objetivos desde o início do desenvolvimento deste trabalho foi de analisar qual é a importância da flexibilidade dos receptores em docagem molecular. Para isso, foi proposto um ambiente que permitisse um estudo sobre como ocorrem as interações receptor-ligante em simulações de docagem com o modelo FFR. Dessa forma, para alcançar esse objetivo, decidiu-se utilizar nas análises as distâncias mínimas entre os átomos do ligante e os átomos dos resíduos do receptor e

o valor de FEB de cada experimento de docagem molecular executado. Ao contrário do trabalho descrito no final do Capítulo 3 [MAC08b, MAC08a], onde os atributos eram indicações se havia ou não interação entre determinado resíduo do receptor e o ligante, nos arquivos de entrada a partir desse momento são consideradas os valores das distâncias.

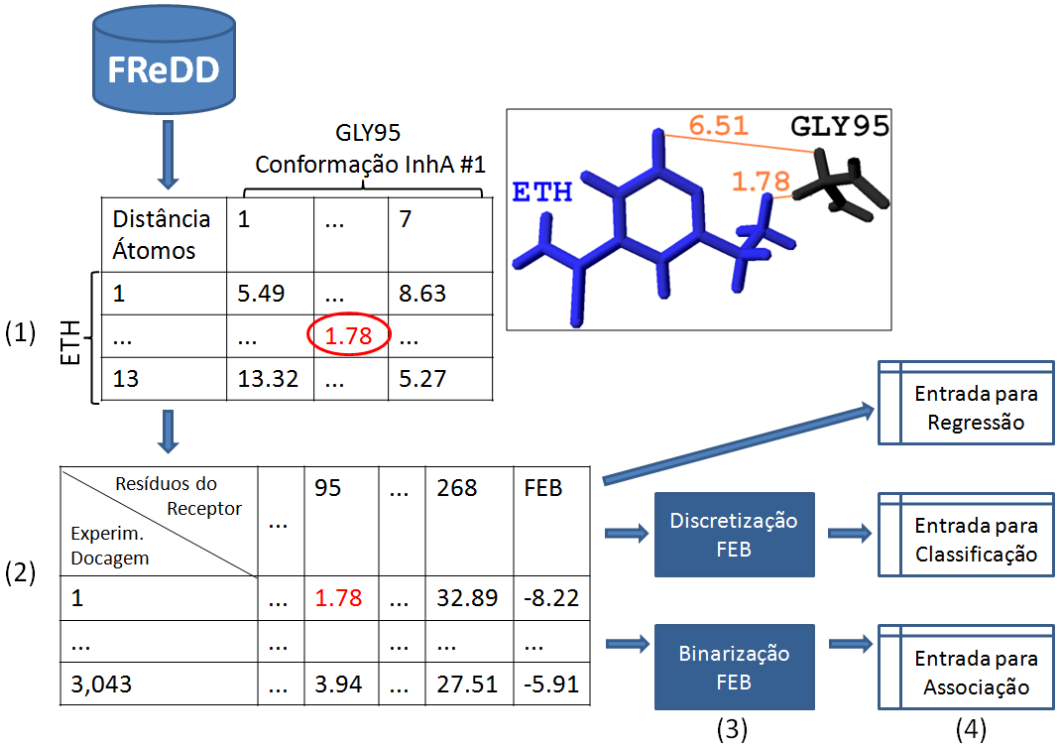


Figura 5.6: Etapas de pré-processamento dos dados do FReDD para a geração das entradas para os algoritmos de mineração de dados. (1) Definição de cada atributo do arquivo de entrada. (2) Um exemplo de arquivo de entrada do complexo InhA-ETH. (3) Passos intermediários no pré-processamento necessários para a aplicação de algumas técnicas de mineração de dados. (4) Os arquivos de entrada para cada uma das técnicas de mineração aplicadas.

Para obter as distâncias mínimas (em $\mathring{A}$) entre todos os átomos do ligante em sua posição final após a docagem e os átomos dos resíduos do receptor, que serão então os atributos preditivos nos arquivos de entrada, é preciso combinar os campos das tabelas *Coord_Atom_Prot* e *Coord_Atom_Lig* de forma a obter uma tabela de distâncias como a mostrada na Figura 5.6(1). Nesse exemplo é considerado o ligante ETH, com seus 13 átomos após a preparação para a docagem molecular (linhas da tabela de exemplo) e os 7 átomos presentes no resíduo do receptor GLY95 (Glicina 95), colunas na tabela de exemplo, cuja combinação totaliza 91 distâncias a serem calculadas para verificar qual é a distância mínima (marcada em vermelho na Figura). Ao lado da tabela, a ilustração desse ligante mostra algumas dessas distâncias calculadas.

5.4.2 (2) Geração dos Arquivos de Entrada ARFF

A partir das distâncias mínimas calculadas na etapa (1) do pré-processamento, é necessário combiná-las em um arquivo de entrada único que contém as distâncias mínimas entre todos os

resíduos do receptor e o ligante para cada uma das conformações do ligante (que podem ser todos os resultados de docagem ou somente aqueles *runs* de melhor FEB). Para a obtenção desta é necessário que sejam efetuados 3.024.112 cálculos de distâncias mínimas para o NADH (268 atributos de distância * 11.284 instâncias), 8.152.560 para o PIF, 7.603.160 para o TCL e 8.155.240 para a ETH. Esse arquivo único está exemplificado em parte na Figura 5.6(2) para o ETH.

Inicialmente tentou-se gerar essa tabela a partir da execução de uma única consulta SQL que calculasse todas essas distâncias para todos os ligantes ao mesmo tempo. Porém, essa consulta não foi processada pelo PostGreSQL, uma vez que, envolvia o cálculo em memória de quase 27 milhões distâncias mínimas (considerando os 4 ligantes). Na tentativa de que as instâncias da tabela fossem obtidas a partir de uma única consulta SQL buscou-se otimizar o FReDD, criando diferentes índices e tabelas intermediárias, porém nenhuma das consultas executadas dessa forma obteve sucesso já que o aplicativo PGAdim (que administra BD desenvolvidos para o PostGreSQL) parava sua execução após, em média, 48 horas ininterruptas.

Devido a esse problema, optou-se pelo desenvolvimento de um *script Python* que conectasse ao FReDD executando uma sequência de consultas, considerando em cada uma, uma conformação e um resultado de docagem a ser analisado e escrevendo os resultados em um arquivo de saída para cada ligante, ao final de cada consulta. Durante a execução das consultas são utilizadas tabelas intermediárias criadas para otimizá-las (com o uso das tabelas intermediárias houve uma redução no tempo de execução de cada consulta de em torno de 70%). A tabela *Conf_docking_1ENY* reune as informações de todos os átomos de todos os resíduos de todas as conformações do receptor InhA (Código PDB:1ENY). Assim, a consulta nesses dados não é mais executada a partir de junções de várias tabelas mas diretamente acessando os registros desta tabela intermediária. As outras tabelas intermediárias foram *Conf_docking_NADH_1ENY*, *Conf_docking_ETH_1ENY*, *Conf_docking_TCL_1ENY* e *Conf_docking_PIF_1ENY*. Essas tabelas armazenam os registros de coordenadas de todos os átomos de todas conformações do respectivo ligante. Assim, a consulta nos dados de coordenadas dos ligantes é feita diretamente nessas tabelas, não sendo necessária a junção de várias outras, o que envolveria um custo computacional bem maior.

O algoritmo a seguir resume o *script Python* escrito para gerar essas entradas. Para facilitar o entendimento do algoritmo, serão utilizadas variáveis e não os nomes das tabelas e os nomes dos campos. Sendo:

- $TotalConformacoes_R$: corresponde ao número total de conformações do receptor;
- $TotalConformacoes_L$: corresponde ao número total de conformações de um determinado Ligante, considerando todos os *runs*;
- $TotalResiduos_R$: armazena o número total de resíduos do receptor;
- $Residuo_R$: é um dado resíduo do receptor;
- $TotalAtomosResiduo_R$: armazena o total de átomos do $Residuo_R$;

- x_R, y_R e z_R: correspondem as coordenadas espaciais de cada átomo de determinado $Residuo_R$;
- $Ligante_L$ é um determinado ligante;
- $TotalAtomoLigante_L$: armazena o total de átomos do $Ligante_L$;
- x_L, y_L e z_L: correspondem as coordenadas espaciais de cada átomo de determinado $Ligante_L$;
- $DistM_{i,j}$: matriz que armazena todas as distâncias (tabela exemplo da Figura 5.6(1));
- $ResultM_{t,r}$: matriz que armazena todos os resultados (Figura 5.6(1));
- $MinimaDist$: armazena o valor mínimo da matriz $DistM_{i,j}$.

Algoritmo 5.1: Algoritmo de cálculo de distâncias entre átomos do ligante e átomos de resíduos do receptor. Adaptado de [WIN10a].

```
1: MinimaDist = 1000
2: Para w = 1 até TotalConformacoes_R
3:     Para t = 1 até TotalConformacoes_L
4:         Para r = 1 até TotalResiduos_R
5:             Para i = 1 até TotalAtomosResiduo_R
6:                 Para j = 1 até TotalAtomoLigante_L
7:                     DistM_{i,j}[i,j] = sqrt((x_Ri - x_Lj)^2 + (y_Ri - y_Lj)^2 + (z_Ri - z_Lj)^2)
8:                     Se DistM_{i,j}[i,j] < MinimaDist então
9:                         MinimaDist = DistM_{i,j}[i,j]
10:                    Fim Se
11:                Fim Para
12:            Fim Para
13:        ResultM_{t,r} = MinimaDist
14:        Fim Para
15:    Fim Para
16: Fim Para
```

Durante essa etapa de preparação dos dados foram então gerados os arquivos ARFF (*Attribute-Relation File Format*) para serem utilizados no WEKA e cujo conteúdo corresponde à matriz $ResultM_{t,r}$. Um arquivo ARFF é composto por duas seções distintas: um cabeçalho que contém o nome da relação, a lista de atributos e seus tipos, e a seção de dados, onde cada linha corresponde a uma instância e os atributos são delimitados por vírgulas e devem aparecer na ordem com que foram declarados no cabeçalho [WIT05], conforme mostra o exemplo na Figura 5.7.

5.4.3 (3) Preparação dos arquivos ARFF para as técnicas de mineração

A base do ARFF mostrado em parte na Figura 5.7 é a da Figura 5.6(2). Todos os atributos preditivos do arquivo ARFF de entrada são as distâncias mínimas e o atributo-alvo é a FEB. Porém esses arquivos ARFF, para algumas técnicas de mineração de dados ainda precisam ser modificados,

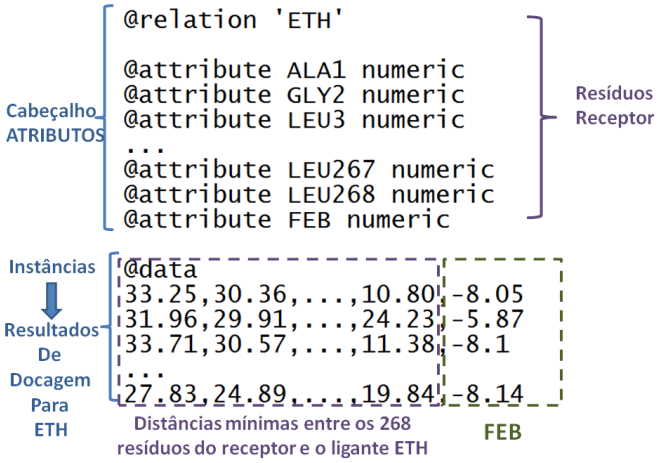


Figura 5.7: Exemplo de arquivo ARFF.

conforme mostra a Figura 5.6(3). Para os experimentos com regressão utilizando árvores modelo, o ARFF da Figura 5.7 está pronto, pois o atributo-alvo para esse tipo de técnica é um valor numérico. Entretanto, para os experimentos com classificação aplicando árvores de decisão é necessário discretizar o valor da FEB, pois para esses experimentos o atributo-alvo deve ser categórico (Capítulo 6). Além do mais, para experimentos com Associação e para análises sobre as interações entre o receptor-ligante, onde não há atributo-alvo, os atributos preditivos precisam ser binarizados, onde:

$$Atributo = \begin{cases} 1 & \text{se} \quad Distancia_{RL} \leq 4{,}0\text{Å} \\ 0 & \text{se} \quad Distancia_{RL} > 4{,}0\text{Å} \end{cases}$$

Onde:$Distancia_{RL}$ = distância mínima entre um determinado resíduo do receptor e um ligante. O valor de 4,0 Å foi determinado de forma a estar associado a categorização de distâncias entre átomos doadores-aceitadores em ligações de hidrogênio e são classificadas em [JEF97, SIL09]:

- **Forte** - equivale a distâncias entre os átomos de 1,9 Å a 2,5 Å;
- **Moderada** - distâncias entre os átomos de 2,5 Å a 3,2 Å;
- **Fraca** - distâncias entre 3,2 Å e 4,0 Å;

Distâncias maiores do que 4,0 Å indicam que determinado resíduo não estabelece nenhum contato com nenhum átomo do ligante em determinado resultado de docagem.

5.5 Análises preliminares com o FReDD

Um dos objetivos desse trabalho é investigar a importância da flexibilidade do receptor em suas interações intermoleculares com pequenas moléculas ou ligantes. Para isso, a partir dos arquivos de entrada de distâncias entre resíduos do receptor e ligantes, foram realizadas análise preliminares nos dados com o objetivo de verificar quais são os resíduos que permanecem em contato (ou seja, que tem como atributo no arquivo o valor 1) na maior parte de todas as execuções de cada simulação de

docagem molecular com o FFR para cada um dos ligantes. O resultado que descreve os 10 resíduos que mais interagem (Top 10) com cada um dos 4 ligantes, mostrados na Figura 5.8.

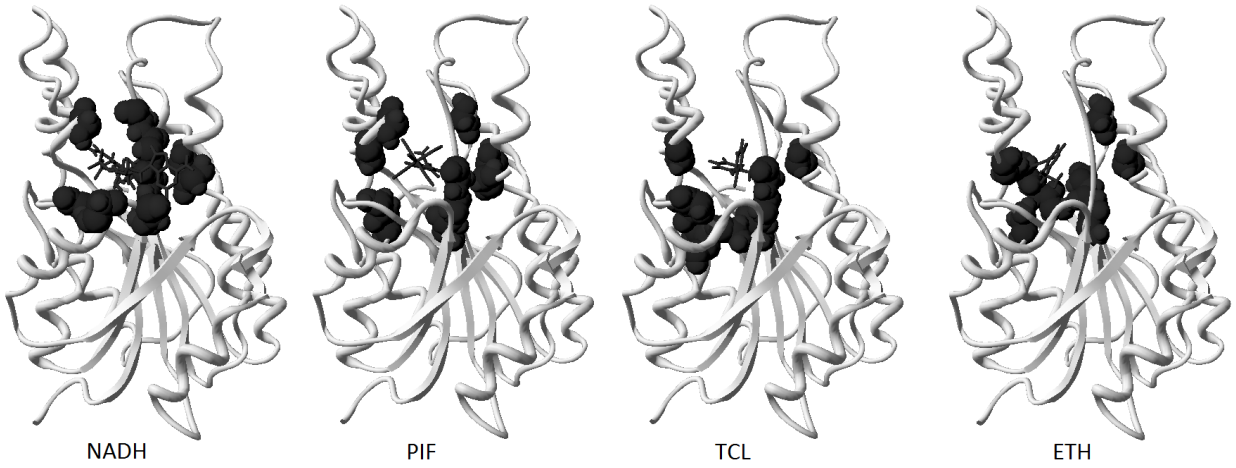


Figura 5.8: Resíduos do receptor Top 10, os 10 resíduos que mais interagem com cada um dos ligantes. Em cinza em *Ribbons* uma estrutura do receptor. Os Top 10 resíduos de cada ligante são apresentados na forma de esferas de *van der Walls* e o ligante na forma de palitos.

A união dos Top 10 de cada ligante é a lista final de 25 resíduos descrita na Tabela 5.3 onde as células em destaque indicam os resíduos que mais interagiram com cada um dos ligantes.

A Tabela 5.4 apresenta esse total de resíduos do receptor que interagem com o modelo FFR (coluna 2 na tabela) em pelo menos um dos resultados e os compara com o total de resíduos que interagem com cada ligante em um resultado de docagem com a estrutura cristalográfica do receptor InhA, o qual chamamos de modelo rígido do receptor (RR - *Rigid Receptor*), análise realizada por uma inspeção visual com um programa visualizador de estruturas de macromoléculas (coluna 3 na tabela) [GUE97]. Além disso, na coluna 4 da tabela é apresenta quantos resíduos do modelo RR são comuns a seleção dos Top 10 para cada ligante [WIN10a].

A Tabela 5.4 confirma que é muito importante considerar a flexibilidade em simulações de docagem molecular. Por exemplo, para o NADH, somente 22 resíduos do modelo RR interage com este ligante. Quando sua flexibilidade é considerada, existem 185 resíduos interagindo em pelo menos um *run* com o modelo FFR. Para esse ligante, 9 dos 10 Top 10 aparecem na seleção do modelo RR. Isto acontece porque o NADH é o ligante natural e sua região de ligação nesse receptor é bem conhecida mas ainda assim a flexibilidade apresenta um papel muito importante em mediar o estado de equilíbrio do complexo [WIN10a]. Para os ligantes PIF e TCL, 7 dos Top 10 são os mesmos (Tabela 5.4). Isto acontece porque a região de ligação para esses 2 ligantes é a mesma, próxima a onde liga-se o substrato. Para o TCL, 139 resíduos do modelo FFR interagem com o ligante enquanto que somente 12 do modelo RR, e destes, não mais do que 5 aparecem na lista de Top 10 para esse ligante. Isto significa que há outros 5 resíduos que interagem muitas vezes no modelo FFR que não aparecem na seleção RR [WIN10a]. Como os ligantes PIF e TCL tem quase o dobro do tamanho do ligante ETH, esperava-se que os mesmos não interagissem na mesma região do receptor, o que é indicado pelos 7 dos Top 10 resíduos serem diferentes para o ETH.

Tabela 5.3: Para cada ligante foram selecionados 10 resíduos e a união dos 10 de cada ligante resultou nos 25 resíduos do receptor descritos nesta tabela.

Residue	ETH	NADH	PIF	TCL
ALA21	3.112	7.138	8.414	15.252
ALA190	23.480	3.744	13.714	7.861
ALA197	1.868	14.127	26.114	6.527
ARG42	120	13.959	4.716	1.940
ASP147	22.645	6.795	10.848	9.585
GLY13	3.647	13.479	15.500	19.900
GLY95	5.521	20.288	27.561	23.852
GLY191	22.909	2.162	13.837	839
ILE15	2.079	17.839	13.226	20.397
ILE20	25.480	11.735	23.312	23.393
ILE94	7.570	17.363	26.632	24.460
ILE121	161	15.782	1.430	10.431
ILE193	23.023	6.005	15.519	1.617
LYS164	24.658	14.627	21.821	12.887
MET97	660	14.153	16.661	1.241
MET146	25.368	10.858	18.352	12.625
MET160	21.653	12.355	20.681	6.375
PHE40	446	15.864	4.823	11.220
PHE96	1.355	20.520	19.401	9.292
PHE148	25.961	8.498	15.772	9.923
PRO192	22.816	3.825	13.968	1.240
SER19	3.532	12.619	26.490	23.659
SER93	12.580	12.957	21.726	24.319
SER122	2.421	12.335	19.805	3.111
THR195	17.601	12.348	26.353	20.474

Tabela 5.4: Análises de interações intermoleculares entre modelo FFR-ligantes e modelo RR-ligantes.

Ligante	Interações FFR-Ligante	Interações RR-ligante	RR ∩ Top 10
NADH	185	22	9
PIF	165	13	8
TCL	139	12	5
ETH	105	8	4

5.6 Considerações Finais

Esse capítulo descreveu o banco de dados FReDD e todas as suas tabelas. Para cada tabela foram descritos os principais campos, como que seu conteúdo foi obtido, de que forma os dados foram armazenados e o total de registros. Uma grande parte dos dados armazenados no FReDD foram utilizados nos experimentos de Mineração de Dados que serão descritos nos próximos capítulos. Até o momento não foi encontrado nenhuma base de dados que tenha sido desenvolvida com o mesmo propósito do FReDD. Há uma plataforma de integração de dados para Triagem Virtual que

é detalhado no Capítulo 9.4 [COC10], mas esta não armazena informações como resultados de docagem molecular nem conformações de uma trajetória de DM.

Nas seções que descrevem o pré-processamento é apresentada uma metodologia de preparação dos dados de docagem molecular para a geração de entradas para mineração de dados pode ser aplicada a diferentes complexos com o objetivo de descoberta de conhecimento sobre as interações FFR-ligante.

Além do mais, as análises preliminares sobre os dados armazenados no FReDD mostram informações que não seriam obtidas se o modelo rígido do receptor fosse considerado ou sem uma preparação apropriada de todos os dados resultantes de simulações de docagem molecular com o modelo FFR. Não foi encontrado até o momento trabalhos que tenham investigado a flexibilidade dos receptores em docagem molecular da forma como apresentado nesta Tese. Essas análises descritas na Seção 5.5 podem ser muito importantes tanto para o entendimento de como é a interação do receptor InhA com ligantes, quanto para a busca de novos inibidores para essa enzima. Nesse sentido, já esta em finalização um trabalho de mestrado desenvolvido pelo aluno Christian Quevedo no LABIO-GPIN para a busca de ligantes considerando informações da trajetória de receptores.

No próximo capítulo são descritos os experimentos de classificação com árvores de decisão, realizados também para entendimento das simulações de docagem molecular com o FFR, de forma a determinar a relação dos resíduos do receptor em contato com o ligante com o valor de FEB.

6. RESULTADOS 2 - APLICAÇÃO DE CLASSIFICAÇÃO POR ÁRVORES DE DECISÃO

Neste capítulo serão apresentados os resultados obtidos com a aplicação de classificação com árvores de decisão utilizando o algoritmo J48 do WEKA [HAL09]. Uma das principais contribuições desse capítulo é a metodologia proposta de discretização do atributo-alvo, a FEB, que se utiliza dos valores de moda e desvio padrão da mesma. Essa metodologia proposta é então comparada com 2 métodos de discretização clássicos, por frequência e por intervalos de tamanho igual. A comparação dos métodos é feita com base no impacto dos mesmos no resultado das árvores de decisão geradas. Dessa forma é possível indicarmos, ao final desse capítulo, qual é o método de discretização que mais se aplica a esse tipo de dados de docagem molecular. Além do mais, a partir das árvores geradas nós apresentamos uma outra metodologia para análise das interações FFR-ligante, diferente da apresentada nas análises preliminares do Capítulo 5.

Os resultados apresentados nesse capítulo estão publicados:

- como resumo expandido no LNBI-LNCS [MAC10c] durante o evento *Brazilian Symposium on Bioinformatics* de 2010;
- como artigo na conferência *IADIS International Conference Applied Computing* de 2010 [MAC10b];
- como capítulo do livro *Tópicos em sistemas colaborativos, multimídia, web e banco de dados.* de 2010 [WIN10b]. Nesse capítulo de livro é apresentado um exemplo de utilização de árvores de decisão para o NADH;
- como uma parte do artigo [MAC11b] que está na 3°rodada de revisão para publicação no *WIREs Data Mining and Knowledge Discovery* que consiste em um resumo de todos os experimentos de mineração realizados durante o desenvolvimento desta Tese.

O principal objetivo da execução de experimentos de mineração de dados, no contexto desse trabalho, é a obtenção de modelos que, de alguma forma, selecionam quais as conformações do receptor se mostram mais promissoras, de forma que futuros experimentos de docagem molecular com diferentes ligantes utilizem não todas as conformações do receptor, mas sim somente aquelas que, para os ligantes analisados nesse trabalho, obtiveram os melhores resultados. Dessa forma, o tempo de execução de cada experimento de docagem molecular considerando um diferente complexo receptor-ligante será reduzido consideravelmente. Nesse sentido, o primeiro trabalho que foi realizado para a geração de modelos que relacionassem os valores de distância entre os resíduos do receptor e o ligante e os valores finais de FEB consistiu na utilização da técnica de mineração de dados Classificação utilizando o WEKA [HAL09] para a execução dos experimentos.

Entre os diferentes algoritmos de classificação, neste trabalho, foi escolhido o algoritmos de classificação por árvores de decisão. De acordo com Freitas et al. [FRE10] a árvore de decisão tem a vantagem de sua saída graficamente representar a descoberta de conhecimento e indicar a importância dos atributos utilizados para predição. E, como este trabalho está inserido em um contexto interdisciplinar, era necessário a escolha de um algoritmo de classificação onde a saída fosse facilmente entendível e não uma caixa preta como outros algoritmos de classificação como *Support Vector Machines* (SVM) ou redes neurais. O WEKA implementa vários algoritmos de classificação baseados em árvores de decisão, com por exemplo o ADTree, NBTree, J48 entre outros. Nesta Tese, optou-se pelo J48, implementação do WEKA para o algoritmo C4.5 [QUI86]. Todos os experimentos com esse algoritmo foram realizados considerando como arquivo de entrada somente o melhor *run* de cada execução de docagem, o que chamamos de experimentos com *Best FEB*, e todos os resultados utilizados são dos experimentos de docagem molecular Fase 1. O resumo dos resultados desse experimento estão descritos na Tabela 3.1 do Capítulo 3.

Conforme descrito no Capítulo 4, na classificação, o atributo-alvo precisa ser uma classe, ou seja, o mesmo não pode ser um valor discreto como é o valor da FEB. Sendo assim, é necessário discretizá-lo, sendo esse um dos passos do pré-processamento dos dados, descrito na próxima seção deste capítulo.

6.1 Discretização do Atributo Classe

Segundo Tan *et al.* [TAN05] quando se decide discretizar um atributo contínuo transformando-o em categórico é necessário se decidir em quantas categorias isso será realizado e em como será feito o mapeamento. O processo de discretização segue principalmente 2 passos:

1. os dados contínuos são organizados de alguma forma, e então divididos em n intervalos especificados por $n-1$ pontos de divisão. ;

2. é definido como todos os valores de um intervalo serão mapeados para um mesmo valor categórico.

Para as abordagens testadas neste trabalho, os dados foram discretizados em $n=2$, $n=3$, $n=4$ e $n=5$ intervalos. Porém os resultados se mostraram mais promissores com 5 classes e devido a isso, são apresentados somente esses resultados utilizando $n=5$: *Excelente*, *Bom*, *Regular*, *Ruim* e *Muito_Ruim*.

Há diferentes abordagens para a realização da discretização, entre as quais nesse trabalho foram aplicadas: discretização por frequência (Método 1), discretização por grupos de mesmo tamanho (Método 2) e discretização por moda e desvio padrão (Método 3). As Figuras 6.1 e 6.2 mostram os histogramas de distribuição do valor de FEB para os 4 ligantes, mostrando o resultado do número de instâncias que permaneceram em cada classe, de acordo com os 3 métodos de discretização aplicados.

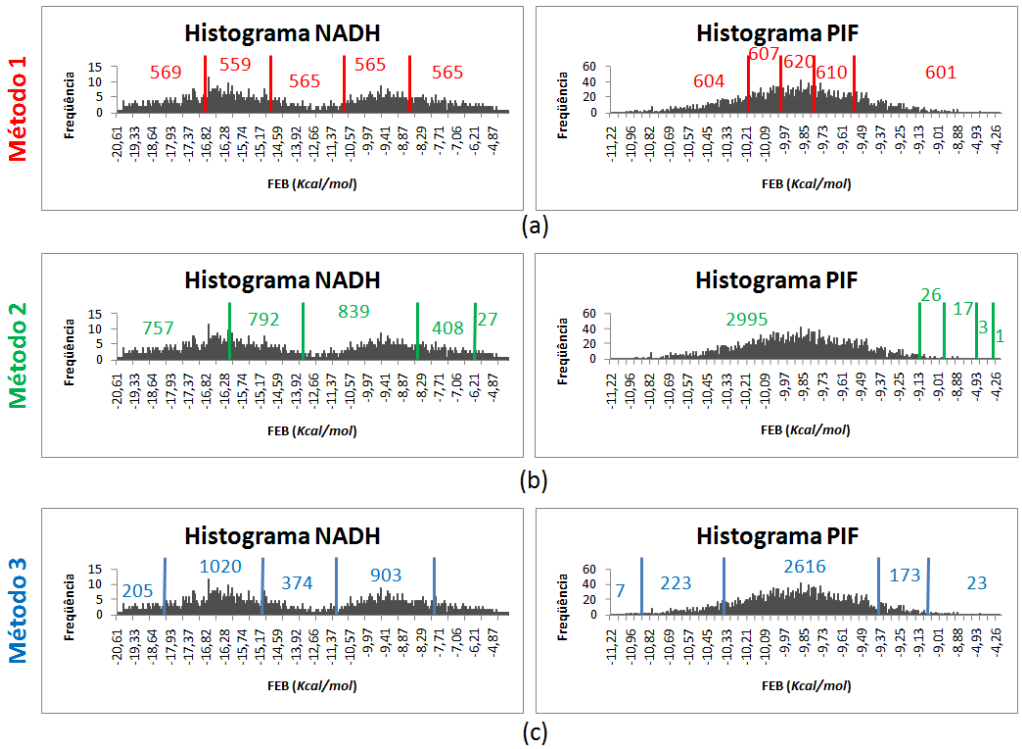


Figura 6.1: Histograma dos ligantes NADH e PIF. Em (a) a discretização pelo Método 1, por frequência. (b) a discretização pelo Método 2, por tamanho igual e (c) discretização utilizando os valores de moda e desvio.

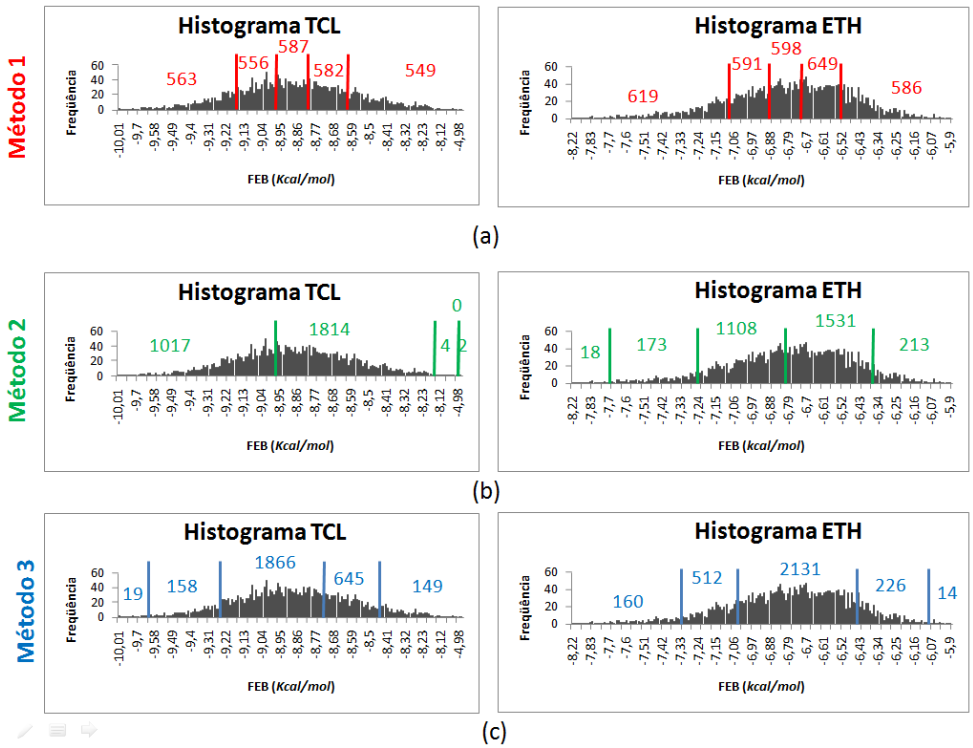


Figura 6.2: Histograma dos ligantes ETH e TCL. Em (a) a discretização pelo Método 1, por frequência. (b) a discretização pelo Método 2, por tamanho igual e (c) discretização utilizando os valores de moda e desvio.

6.1.1 Discretização por Frequência:

Uma das abordagens mais simples é a discretização dos dados em conjuntos de mesmo tamanho [TAN05], onde os dados são divididos em um número específico de intervalos com aproximadamente o mesmo número de instâncias. Então, considerando k o número de intervalos definidos pelo usuário e m o total de instâncias, esse método divide a variável contínua em k intervalos onde cada intervalo contém aproximadamente $\frac{m}{k}$ instâncias [TAN05]. Como resultado, descritos nas Figuras 6.1(a) e 6.2(a), é possível verificar que as classes ficam balanceadas, todas com aproximadamente o mesmo número de instâncias.

6.1.2 Discretização por Tamanho de Intervalo Igual:

De acordo com Dougherty et al. [DOU95] este é o método de discretização mais simples, embora seja vulnerável a *outliers*. Nessa abordagem, para cada atributo discreto a ser discretizado, seus valores são ordenados e então divididos em k intervalos, onde cada intervalo tem o mesmo tamanho. Sendo k o número de intervalos, o tamanho δ dos intervalos de um determinado atributo x é definido por:

$$\delta = \frac{X_{max} - X_{min}}{k} \tag{6.1}$$

O resultado desse método de discretização está descrito nas Figuras 6.1(b) e 6.2(b). É importante ressaltar que nas Figuras não está representada toda a distribuição de FEB, pois para intervalos que continham poucos valores, os mesmos foram condensados pelo programa *Microsoft Excel* (onde foram feitos os histogramas) para melhorar a visualização. Dessa forma, principalmente para os histogramas dos ligantes TCL e PIF, tem-se a impressão de que os intervalos não têm o mesmo tamanho.

Como resultado desse método de discretização, as classes não ficaram balanceadas, em especial para os resultados dos ligantes TCL e PIF, como pode ser observado nas Figuras 6.1(b) e 6.2(b). Considerando os valores da Tabela 3.1 do Capítulo 3 para os resultados *Best FEB*, para o TCL a FEB varia de -10.0 Kcal/mol até -4.9 kcal/mol, porém o valor de Moda é -9.0 kcal/mol, um valor muito próximo do valor de FEB máximo, o que divide então a maioria das instâncias como dos primeiro ou segundo intervalos. O mesmo acontece na distribuição de FEB do ligante PIF, onde o valor da Moda (-9,9 kcal/mol) é muito mais próximo do valor máximo de FEB do que de seu valor mínimo, causando o mesmo efeito que ocorre no ligante TCL.

6.1.3 Discretização utilizando Moda e Desvio Padrão:

Esse método de discretização é proposto nesta Tese e está publicado em [MAC10c, MAC10b]. Ele considera os valores de moda e desvio padrão do atributo que está sendo discretizado. O objetivo deste método de discretização proposto era de que atributos nas extremidades ficassem em um mesmo intervalo, por exemplo, que os melhores valores de FEB permanecessem em um

intervalo e os piores em outro intervalo diferente. Esse método de discretização segue a Equação 6.2 considerando uma discretização em 5 intervalos. Esta equação é dependente do número de classes e deve ser modificada se um número de classes diferente for considerado. Nessa Equação 6.2, FEB é o atributo que está sendo discretizado, M_o é o valor de moda da distribuição do atributo FEB e σ representa o desvio padrão do atributo FEB.

$$Classe = \begin{cases} Excelente & \text{se} & M_o - 2*\sigma & > & FEB & & \\ Bom & \text{se} & M_o - \sigma & > & FEB & \geq & M_o - 2*\sigma \\ Regular & \text{se} & M_o + \sigma & > & FEB & \geq & M_o - \sigma \\ Ruim & \text{se} & M_o + 2*\sigma & > & FEB & \geq & M_o + \sigma \\ Muito\_Ruim & \text{se} & & & FEB & > & M_o + 2*\sigma \end{cases} \quad (6.2)$$

Como se pode verificar nas Figuras 6.1 e 6.2 essa discretização para alguns ligantes é balanceada, porém a maioria das instâncias, para os 4 ligantes, ficou classificada como *Regular*. Entretanto, o objetivo deste método de discretização foi alcançado para os 4 ligantes, permanecendo os melhores valores de FEB em um grupo diferente dos piores valores, para os 4 ligantes.

6.2 Resultados com o Algoritmo J48

Para a execução dos experimentos com o algoritmo J48 do WEKA foram geradas entradas diferentes para os 4 ligantes e para os 3 métodos de discretização, totalizando 12 arquivos de entrada, onde os atributos preditivos são as 268 distâncias mínimas dos resíduos do receptor para cada ligante e o atributo-alvo é a classe de FEB de cada um dos resultados de docagem molecular considerados [MAC10b]. Os resultados dos experimentos utilizando o algoritmo J48 estão descritos na Tabela 6.1.

Para que os modelos gerados fossem mais legíveis, a maioria dos parâmetros do algoritmo J48 permaneceram com seus valores *default*, com exceção do parâmetro $minNumObj$ o qual foi atribuído o valor de 50. Esse parâmetro está relacionado com o número de instâncias mínimo em cada nodo folha. Foram executados experimentos com esse parâmetro com valores de 30, 50, 75 e 100, sendo os melhores resultados obtidos da execução com $minNumObj = 50$. A avaliação dos modelos gerados é feita com a validação cruzada com 10 partições, conforme explicado no Capítulo 4.

Na Tabela 6.1 de resultados tem-se: na primeira coluna a descrição do método de discretização utilizado, na segunda, o nome do ligante, na terceira a Acurácia (Acc.) do conjunto de teste da validação cruzada, na quarta coluna é apresentado o tamanho da árvore final resultante em cada experimento (*Tree Size* - TS). Nas colunas 5 e 6, os percentuais de MAE e RMSE respectivamente. A sétima coluna contém o valor de *F-measure* (FM) de cada modelo gerado. Para detalhes sobre as métricas Acc., TS, MAE, RMSE e FM consultar o Capítulo 4.

Além das métricas já descritas anteriormente, para uma melhor avaliação dos métodos de discretização foi definida a métrica IEGC (*Instances in Excellent or Good Classes*), cujos valores para cada

modelo estão descritos na coluna 8 da Tabela 6.1. Essa métrica calcula o percentual de instâncias que pertencem as classes Excelente ou Bom. Para essa métrica buscamos por valores menores, para que haja uma distribuição mais uniforme das instâncias entre os intervalos de FEB definidos por cada método de discretização [MAC10b].

Tabela 6.1: Resultados dos experimentos utilizando o algoritmo J48 considerando todos os Resíduos.

Método	Ligante	Acc.	TS	MAE	RMSE	FM	IEGC
1	NADH	61,88	61	0,18	0,32	0,62	39,96
1	PIF	31,92	71	0,30	0,40	0,31	39,81
1	TCL	30,49	61	0,30	0,40	0,30	39,44
1	ETH	36,38	77	0,28	0,39	0,35	39,76
2	NADH	73,53	43	0,14	0,28	0,73	54,87
2	PIF	98,68	3	0,01	0,07	0,98	99,31
2	TCL	64,93	49	0,16	0,30	0,64	99,79
2	ETH	61,02	41	0,21	0,33	0,57	06,28
3	NADH	75,41	35	0,13	0,27	0,75	43,39
3	PIF	86,55	5	0,09	0,22	0,81	07,56
3	TCL	66,23	17	0,19	0,31	0,58	06,06
3	ETH	70,32	29	0,17	0,29	0,65	22,08

Considerando o Método 1, por frequência, foram obtidos os piores resultados para todos os ligantes, o que mostra que esse tipo de discretização, para esse tipo de dado, não é apropriado.

O Método 2, por tamanho de intervalo igual, obteve melhores resultados para o PIF. Entretanto para esse ligante, esse método tem 99,31% das instâncias classificadas como Excelente ou Bom (IEGC). Isso significa que o modelo gerado por PIF-Método 2 não é útil para extração de conhecimento sobre os resíduos do receptor envolvidos em bons valores de FEB, pois a maioria das instâncias está classificada em uma mesma categoria. Para o TCL, o método 2 foi melhor em 3 das 5 métricas avaliadas. Entretando, assim como ocorreu com o PIF, para o TCL-Método 2, 99,79% das instâncias estão classificadas como Excelente ou Bom. Dessa forma, se o valor de IEGC for considerado, mesmo com melhores valores de acurácia, esses modelos são distorcidos.

O Método 3, proposto em [MAC10c], obteve melhores resultados em todas as métricas para os ligantes NADH e ETH e em 2 das 5 métricas para o TCL. Embora os valores não sejam melhores para todas as métricas, os modelos gerados com esse método de discretização foram mais legíveis, ou seja, permitem uma melhor interpretação por serem árvores com poucos nodos. Consequentemente, esses modelos são mais aplicáveis, permitindo que mais informação sobre a interação receptor-ligante seja extraída dos modelos gerados.

Com o objetivo de tentar melhorar os modelos gerados por árvore de decisão, decidiu-se executar um segundo conjunto de experimentos de classificação, onde os atributos preditivos dos arquivos de entrada foram selecionados. Essa seleção de atributos foi realizada para eliminar todos os atributos de distâncias mínimas de resíduos que em nenhum resultado de docagem molecular estabeleceram contato com o ligante, ou seja, que a distância mínima considerando todas as simulações de docagem

foi maior do que 4,0 Å [MAC11b]. Assim, o total de atributos (preditivos mais o atributo-alvo) de cada arquivo de entrada para esse segundo conjunto de experimentos é de 70, 81, 66 e 88 para os ligantes NADH, PIF, TCL e ETH respectivamente. Os resultados do segundo conjunto de experimentos com árvores de decisão estão resumidos na Tabela 6.2, cuja descrição das colunas e linhas é a mesma da Tabela 6.1.

Tabela 6.2: Resultados dos experimentos utilizando o algoritmo J48 considerando somente os resíduos com distância mínima menor que 4,0 Å.

Método	Ligante	Acc.	TS	MAE	RMSE	FM	IEGC
1	NADH	62,42	47	0,19	0,32	0,71	39,96
1	PIF	31,16	65	0,30	0,40	0,30	39,81
1	TCL	29,50	65	0,30	0,40	0,28	39,44
1	ETH	35,82	75	0,28	0,39	0,35	39,76
2	NADH	71,96	35	0,14	0,29	0,70	54,87
2	PIF	98,68	3	0,01	0,07	0,98	99,31
2	TCL	65,99	49	0,21	0,32	0,65	99,79
2	ETH	61,98	43	0,20	0,33	0,57	06,28
3	NADH	72,93	31	0,13	0,28	0,72	43,39
3	PIF	86,78	5	0,09	0,22	0,82	07,56
3	TCL	66,09	15	0,19	0,31	0,58	06,06
3	ETH	69,79	27	0,21	0,29	0,65	22,08

Os resultados obtidos com esse segundo conjunto de experimentos com a seleção de atributos foi muito próximo do primeiro conjunto considerando todos os atributos preditivos de distâncias mínimas. A maior diferença ocorreu para o ligante TCL, que para esse segundo conjunto de experimentos obteve melhores resultados com o Método 3. Para o Método 1 foram obtidos os piores modelos. O Método 2 obteve melhores resultados para o PIF, mas assim como para o primeiro conjunto de experimentos, para esse ligante a maioria das instâncias foram classificadas como Excelente ou Bom. O Método 3, utilizando moda e desvio, obteve os melhores resultados para a maioria das métricas para os ligantes NADH, TCL e ETH.

Considerando os resultados para o segundo conjunto de experimentos e com a discretização pelo Método 3, as árvores de decisão geradas são analisadas a seguir. A árvore obtida para o NADH-Método 3 está descrita na Figura 6.3. As árvores para os ligantes PIF, TCL e ETH estão no Apêndice A.

A raiz da árvore de decisão NADH-Método 3 é o resíduo THR 100 (Treonina 100). Como pode-se observar analisando essa árvore de decisão, a distância desse resíduo do receptor para o NADH é determinante para definir se um resultado de docagem obteve bons valores de FEB (classes E, B e Re) ou valores ruins (R e MR). Uma inspeção visual na estrutura cristalográfica desse receptor mostra que esse resíduo não é diretamente relacionado ao sítio ativo desse receptor. A informação de que esse resíduo é importante para a determinação de bons ou ruins valores de FEB não teria sido obtida sem um processo de KDD.

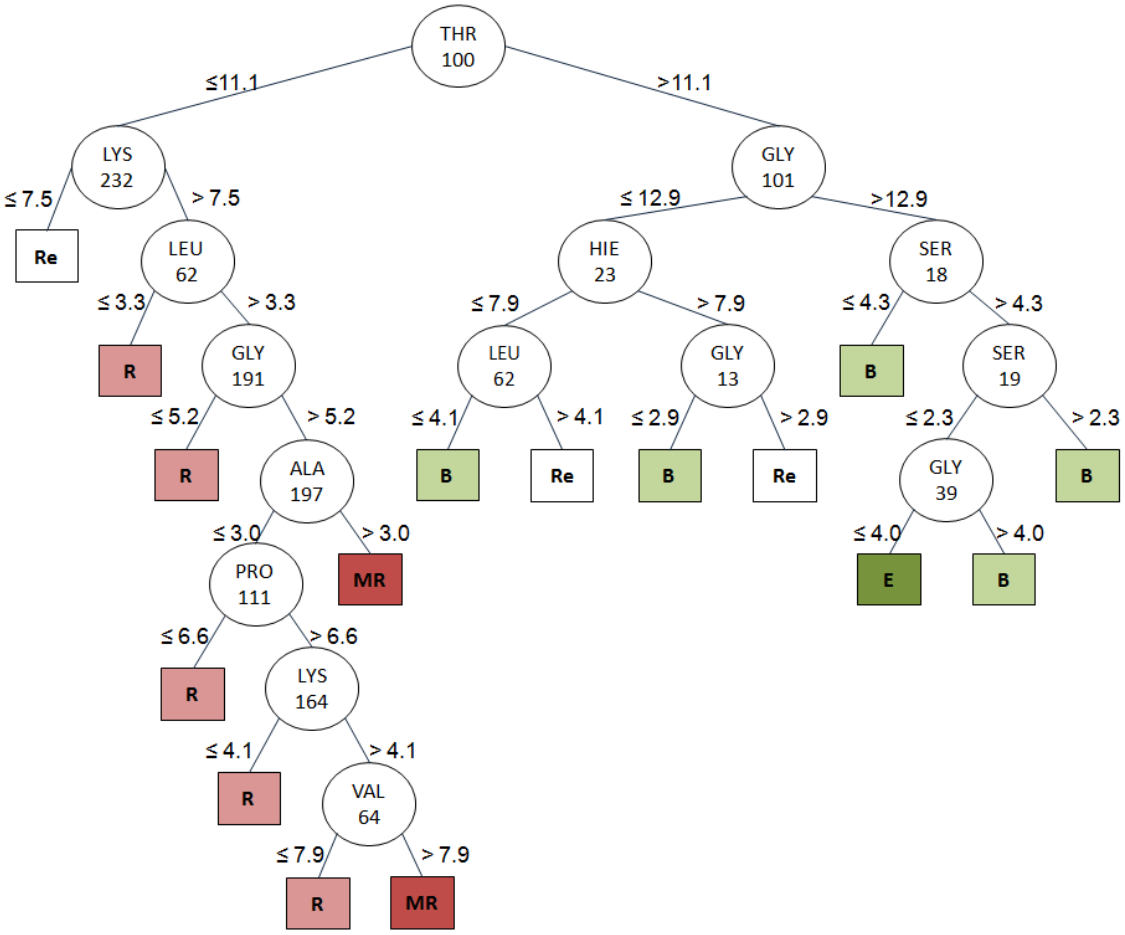


Figura 6.3: Árvore de decisão para o NADH - Método 3. Os nodos-folha estão coloridos de acordo com a classe de FEB obtida pela discretização desse atributo-alvo. Em verde, as classes Excelente e Bom (E e B). Em vermelho, as Classes Ruim e Muito Ruim (R e MR). Em branco, a classe Regular (Re).

Para um melhor entendimento dos modelos gerados com o J48, podem ser extraídas regras de decisão a partir das árvores de decisão [TAN05]. Dessa forma, a partir das árvores para o NADH descrita na Figura 6.3 e para o PIF, TCL e ETH descritas nas Figuras do Apêndice A, pode-se extrair:

- NADH: SE ((THR100 $>$ 11,1 Å) E (GLY101 $>$ 12,9 Å) E (SER18 $>$ 4,3 Å) E (SER19 $<=$2,3 Å) E (GLY39 $<=$4,0 Å)) ENTÃO FEB $=$ EXCELENTE
- PIF: SE (HIE92 $>$ 9,6 Å) ENTÃO FEB $=$ MUITO_RUIM. SE ((HIE92 $<$ 9,6 Å) E (ILE201 $<$ 5,8 Å)) ENTÃO FEB $=$ BOM
- TCL: SE ((PHE96 $>$ 5,4 Å) E (SER93 $>$ 2,3 Å)) ENTÃO FEB $=$ RUIM
- ETH: SE ((ILE14 $>$ 13,4 Å) E (ALA 153 $<=$ 3,0 Å)) ENTÃO FEB $=$ EXCELENTE

6.3 Considerações Finais

Esse capítulo apresentou os experimentos de classificação com árvores de decisão executados durante o desenvolvimento desta Tese. Para a utilização dessa técnica de mineração de dados foi necessária a discretização do atributo-alvo FEB. Foram comparados 3 métodos de discretização, por frequência (Método 1), por intervalos de tamanho igual (Método 2) e o método proposto utilizando moda e desvio padrão da distribuição de FEB dos resultados de docagem para os 4 ligantes (Método 3). A comparação entre os métodos de discretização foi feita baseada na execução de dois conjuntos de experimentos: no primeiro conjunto foram utilizados todos os atributos preditivos de distâncias mínimas entre os resíduos do receptor e os ligantes, no segundo conjunto foi aplicada uma seleção de atributos onde foram excluídos todos os atributos de distâncias mínimas onde o valor para todas as instâncias era maior do que 4,0 $\mathring{A}$. Os resultados para os 2 conjuntos de experimentos foram aproximados: para a maioria das métricas de avaliação o Método 1 apresentou resultados ruins, o Método 2 foi o melhor para o ligante PIF e o Método 3 se mostrou o mais eficiente para os ligantes TCL, NADH e ETH.

Dessa forma, baseado nos resultados apresentados, o método de discretização que se mostrou mais apropriado para ser utilizado em resultados de docagem molecular foi o Método 3, que se utiliza dos valores de média e desvio da distribuição de FEB. Além do método de discretização proposto, a análise dos modelos induzidos obtidos da execução do algoritmo J48 do WEKA é uma outra contribuição deste trabalho, onde uma nova forma de análise da interação receptor-ligante e suas relações com os valores de FEB é apresentada.

Apesar de obter modelos interessantes, e permitir que fossem extraídos conhecimentos sobre a interação receptor-ligante, a utilização das árvores de decisão para a seleção direta de conformações do receptor para utilização em simulações de docagem com ligantes diferentes não é possível de ser feita diretamente. Isso ocorre porque as conformações do receptor com melhor FEB são diferentes para os 4 ligantes, não sendo possível selecionar um conjunto único de conformações mais promissoras. Além do mais, devido aos resultados obtidos não serem promissores para todos os ligantes, acrescido da percepção de que, por causa da variação de FEB ser muito sutil, a determinação de que uma instância pertencia a uma classe ou a outra era determinada por uma diferença de apenas $0,1$ *kcal/mol*, optou-se pela busca de alternativas para o algoritmo J48.

Assim, para prosseguir a pesquisa e a busca de modelos que indicassem características importantes para serem utilizadas na seleção de conformações do receptor, se tornou necessário optar pelo uso de outro algoritmo que não necessitasse que o atributo-classe fosse discretizado, aceitando o valor real da FEB. O algoritmo encontrado com tais características foi o algoritmo de regressão M5P.

7. RESULTADOS 3 - APLICAÇÃO DE REGRESSÃO POR ÁRVORES MODELO

Nesse capítulo são descritos os experimentos realizados com a técnica de mineração de dados de regressão por árvores modelo, utilizando o algoritmo M5P do WEKA. As principais contribuições desse capítulo estão relacionadas à aplicação de regressão para esse tipo de dado de docagem molecular e a comparação dos resultados obtidos com árvores modelo para diferentes formas de preprocessamento desses dados. Os resultados obtidos com o desenvolvimento deste trabalho estão nas seguintes publicações:

Os resultados apresentados nesse capítulo estão publicados:

- como artigo completo publicado no periódico BMC Genomics em 2010 [MAC10a];
- como artigo completo submetido para o periódico *International Journal of Data Mining and Bioinformatics* [WIN11];
- como 2 resumos publicados e apresentados durante o evento ISCB-Latin America [MAC10d, WIN10c] em 2010;
- como parte do capítulo do livro *Tópicos em sistemas colaborativos, multimídia, web e banco de dados* de 2010 [WIN10b]. Nesse capítulo de livro é apresentado um exemplo de utilização de árvores modelo para o NADH;
- como uma parte do artigo [MAC11b] que está na 3° rodada de revisão para publicação no *WIREs Data Mining and Knowledge Discovery* que consiste em um resumo de todos os experimentos de mineração realizados durante o desenvolvimento desta Tese.

Como todos os atributos são valores numéricos e de acordo com Han e Kamber [HAN06] a abordagem mais abrangente para a predição de valores numéricos é a regressão, sendo então este um dos motivos de termos explorado essa técnica de mineração de dados. Além disso, como temos interesse em entender os modelos gerados devido a importância de conhecermos a relação entre a menor distância receptor-ligante e o valor da FEB, estes modelos devem ser tão compreensíveis quanto possível. Desconsiderando que não há um consenso na literatura sobre mineração de dados sobre qual a tarefa de mineração que fornece o resultado mais compreensível, há um acordo de que representações como árvores de decisão e conjunto de regras são melhores de entender do que uma "caixa preta"como o resultado de SVM (*Support Vector Machine*) ou redes neurais [FRE10].

Conforme descrito no Capítulo 4, para a predição numérica há dois tipos de árvores: árvores de regressão e árvores modelo. A principal diferença entre ambas é sobre o conteúdo dos nodos-folha. Cada nodo-folha em uma árvore de regressão armazena um valor contínuo que corresponde a média do valor do atributo predito para as tuplas de teste enquanto que os nodos-folha nas árvores modelo

contém modelos de regressão - uma equação com múltiplas variáveis [HAN06]. Devido ao tipo de resultado apresentado nos nodos-folha optamos por utilizar Árvores modelo (do inglês, *Model Trees*).

O algoritmo de árvore modelo utilizado neste trabalho foi o M5P [WAN97] disponível no pacote WEKA [WIT05]. Como resultado da execução do algoritmo M5P tem-se uma árvore modelo e um conjunto de *Linear Models* (LMs). Cada nodo da árvore modelo é um resíduo e um valor de distância determina a rota a percorrer na árvore. Os nodos folhas todos são LMs. Os LMs são equações que, a partir de uma séria de termos, cada um com seu coeficiente, mais um valor constante, determinam o valor do atributo-classe. Todos os nodos da árvore modelo terminam em um LM. No contexto desse trabalho, os LMs são equações que determinam o valor da FEB a partir de um conjunto de resíduos e seus coeficientes somados a um valor constante.

A aplicação do algoritmo M5P, para todos os ligantes, é realizada considerando todos os *runs* de cada experimento de docagem molecular da Fase 1. Dessa forma, o total de instâncias de cada arquivo de entrada para o algoritmo M5P é de 11.284, 30.420, 28.370 e 30.430 para o NADH, PIF, TCL e ETH respectivamente (o resumo dos resultados dos experimentos de docagem molecular Fase 1 para os 4 ligantes estão descritos na Tabela 3.1 do Capítulo 3).

7.1 Pré-processamento

O primeiro trabalho desenvolvido utilizando o algoritmo M5P [WAN97] está descrito em [WIN10c, WIN11]. Neste trabalho, a partir dos arquivos de entrada que contém os 268 atributos de distâncias mínimas entre os 268 resíduos do receptor e cada um dos ligantes e o atributo-alvo FEB, foram gerados novos arquivos de entrada a partir de diferentes estratégias de seleção de atributos.

7.1.1 Seleção de Atributos Baseada no Contexto

A primeira estratégia de seleção de atributos é proposta em [WIN10c, WIN11] e é baseada no contexto. É a mesma estratégia utilizada nos experimentos de classificação descritos do Capítulo 6. De acordo com Jeffrey [JEF97] a maior distância entre átomos que permite um contato é de 4,0 Å. Assim, essa estratégia de seleção de atributos baseada no contexto adota que distâncias maiores do que 4,0 Å correspondem a resíduos do receptor que não estabelecem contato com nenhum átomo do ligante em nenhuma das simulações de docagem consideradas, e podem, por esse motivo, serem removidos dos arquivos de entrada. Sendo assim, ao aplicarmos essa estratégia de seleção de atributos baseada no contexto os arquivos de entrada permaneceram com o total de atributos descritos na segunda coluna da Tabela 7.1.

7.1.2 Seleção de Atributos com o Algoritmo CFS

Para a comparação do método de seleção de atributos baseado no contexto proposto em [WIN10c, WIN11] com métodos clássicos de seleção de atributos, foi gerada uma nova entrada

Tabela 7.1: Número de atributos em cada arquivo de entrada para as diferentes abordagens de seleção de atributos aplicadas.

Ligante	Distância mínima < 4 (Dist4)	CFS	Dist4 $\cup$ CFS
NADH	84	17	93
TCL	106	14	114
PIF	104	16	108
ETH	105	6	111

onde a seleção de atributos foi realizada a partir da execução do algoritmo *Correlation-based Feature Selection* (CFS) [HAL00] disponível no WEKA. A escolha do algoritmo CFS é que o mesmo é muito utilizado para seleção de atributos ao mesmo tempo em que, apesar do WEKA disponibilizar muitos algoritmos de seleção de atributos, este é um dos poucos que podem ser aplicados a dados de tarefas de regressão. O CFS é um método de seleção de atributos baseado em uma função heurística de avaliação da correlação entre os atributos. O CFS procura por subconjuntos de atributos que contenham características altamente correlacionadas com o atributo-alvo e não correlacionada com os demais atributos. O total de atributos final, após a aplicação do CFS, está descrito na coluna 3 da Tabela 7.1. O CFS é baseado na Equação abaixo:

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{7.1}$$

Onde: M_S é a heurística de um subconjunto S que contém k atributos; $\overline{r_{cf}}$ é a média de correlação entre determinado atributo f e o atributo-alvo; $\overline{r_{ff}}$ é a média de correlação entre 2 atributos.

7.1.3 Seleção de Atributos Combinando Contexto e CFS

Além das duas abordagens descritas acima para seleção de atributos, foi gerada uma terceira entrada que combina os atributos da seleção de atributos baseada no contexto e com o CFS. O resultado dessa terceira abordagem de seleção de atributos está descrita na coluna 4 da Tabela 7.1.

7.2 Primeiro Conjunto de Experimentos Utilizando o Algoritmo M5P

Foram executados experimentos de regressão com o M5P considerando 4 diferente entradas para cada ligante:

1. o arquivo de entrada inicial, com 268 atributos de distâncias mínimas mais o atributo-alvo, a FEB;

2. o arquivo de entrada após a aplicação da seleção de atributos baseada no contexto (Coluna 2, Tabela 7.1);

3. o arquivo de entrada após a aplicação do algoritmo CFS para seleção de atributos (Coluna 3, Tabela 7.1); e

4. o arquivo de entrada com os atributos resultantes da união dos atributos da seleção baseada no contexto e da aplicação do CFS.

O algoritmo M5P tem uma série de parâmetros, dentre eles, concentrou-se em encontrar um bom valor para o parâmetro *Minimum number of instances*, que é relacionado com o tamanho da árvore modelo e com o número de LMs (*Linear Modes*). O valor que utilizou-se nos experimentos foi de 1.000, pois com esse valor foram obtidas árvores modelo legíveis ao mesmo tempo que não tão pequenas que não fornecessem nenhuma informação relevante. Tentou-se a execução do M5P com valores para *Minimum number of instances* de 10, o que gerou árvores impossíveis de serem compreendidas, uma vez que continham milhares de nodos e LMS, 100, que também gerou árvores muito grandes, 1.000, o valor escolhido, 2.000 e 3.000 que começaram a gerar árvores muito pequenas e por isso com pouca informação. É importante mencionar que nos experimentos descritos a seguir eliminou-se todas as instâncias cujo valor de FEB era positivo, uma vez que considerou-se essas instâncias como *outliers*.

Para a avaliação das árvores modelo geradas com o M5P são utilizadas as métricas descritas no Capítulo 4: MAE, RMSE e Correlação, que são calculadas durante a geração dos modelos. Além destas métricas, no trabalho [WIN11] é proposta uma metodologia para avaliação dos modelos gerados baseado também no contexto, mas que avalia o conteúdo das árvores modelos induzidas. A Figura 7.1 mostra a árvore modelo obtida para o arquivo de entrada do ligante NADH com o método de seleção de atributos baseado no contexto. Essa árvore contém 6 nodos-folha, que correspondem a 6 LMS como o da Equação 7.2, que descreve o sexto modelo linear (LM6).

$$\begin{aligned} FEB = &-0.0009 * SER12 + 0.9405 * PHE22 + 0.0013 * THR38 + 0.0035 * ASP63 \\ &+0.0006 * HIE92 + 0.002 * THR100 - 0.5005 * GLY101 - 0.0004 * ALA123 \\ &-0.0015 * ASP147 + 0.0024 * THR161 + 0.0017 * LEU167 + 1.094 * GLY191 \\ &+0.0037 * PRO192 + 0.0015 * ILE193 + 0.0003 * ILE201 - 20.6455 \end{aligned} \quad (7.2)$$

Essa avaliação baseada no contexto tem por objetivo avaliar os modelos de acordo com os resíduos que estão presentes tanto nos nodos internos quanto nos LMs dos nodos-folha, (por exemplo, os resíduos da Figura 7.1 e Equação 7.2, Treonina 100 ($THR100$), Histidina 92 ($HIE92$), Serina 12 ($SER12$), Fenilalanina 22 ($PHE22$), entre outros) para verificar se os mesmos são resíduos dentro do sítio de ligação do receptor, que realmente contribuem para o cálculo da FEB. Para essa análise, um especialista definiu os resíduos do sítio de ligação do receptor, o valor definido como ESR, que corresponde a Seleção de Resíduos do Especialista, que para o receptor InhA totalizou 52 resíduos. Os resíduos que aparecem nas árvores-modelo e nos LMs são definidos como MR (Model-tree Resíduos). Para o cálculo do $F - score$ (Equação 4.5 do Capítulo 4), são utilizados

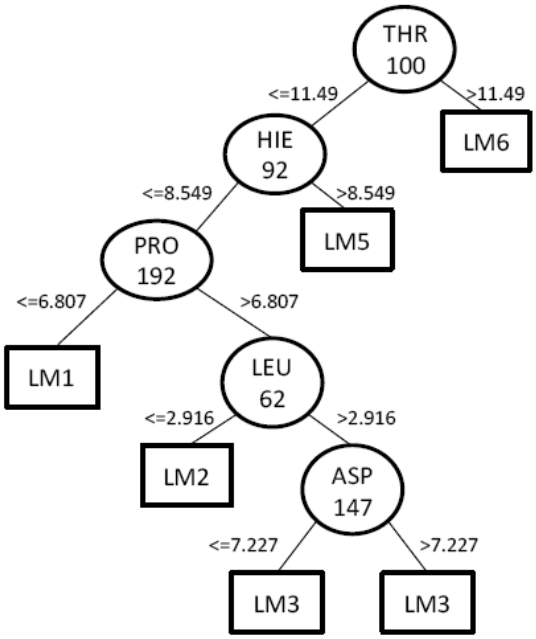


Figura 7.1: Árvore modelo do ligante NADH do primeiro conjunto de experimentos com o M5P, com seleção de atributos baseada no contexto.

as fórmulas de *precisão* e *recall* definidas em Han et al. [HAN06] para avaliação em mineração de texto [WIN11]:

$$precisao, p = \frac{relevantes \cap recuperados}{recuperados} \tag{7.3}$$

$$recall, r = \frac{relevantes \cap recuperados}{relevantes} \tag{7.4}$$

onde, neste contexto, os atributos $relevantes$ são os definidos pelo especialista, ou seja, o valor de ESR e os atributos $recuperados$ correspondem aos atributos MR, que aparecem nas árvores ou nos LMs.

Sendo assim, as Tabelas 7.2 e 7.3 descrevem as métricas de avaliação preditivas e baseadas no contexto respectivamente. Essas métricas são individualmente aplicadas a cada ligante. Na Tabela 7.2 tem-se na primeira coluna os nomes dos ligantes, na segunda coluna a estratégia de pré-processamento utilizada, na terceira coluna o total de nodos da árvore modelo gerada (quanto menor, árvores mais legíveis), na quarta coluna o valor de correlação do conjunto de teste da validação cruzada com 10 partições (valores mais altos indicam modelos melhores), na quinta e sexta colunas os valores de MAE e RMSE (as métricas são detalhadas no Capítulo 4). A Tabela 7.3 contém na primeira coluna os ligantes, na segunda coluna a abordagem de pré-processamento e nas colunas 3, 4 e 5 os valores de precisão, *recall* e *F-score*.

Analisando as Tabelas 7.2 e 7.3 é possível ver que a terceira estratégia de pré-processamento (algoritmo CFS) não é melhor para nenhum dos experimentos realizados. Considerando as métricas preditivas (Tabela 7.2), os arquivos de entrada completo (1) e com seleção de atributos baseado no contexto (2) alternam entre os melhores resultados: para o PIF a estratégia 2 foi melhor para

Tabela 7.2: Avaliação preditiva das árvores-modelo do primeiro conjunto de experimentos com o M5P.

Ligante	Estratégia Pré-processamento	Avaliação			
		Nodos	Correlação	MAE	RMSE
NADH	1	15	0,9536	1,0030	1,3660
	2	5	0,9512	1,0189	1,4000
	3	6	0,9483	1,0578	1,4396
	4	9	0,9513	1,0211	1,3992
PIF	1	22	0,9685	0,3077	0,4071
	2	19	0,9692	0,3053	0,4022
	3	22	0,9653	0,3237	0,4264
	4	19	0,9686	0,3067	0,4060
TCL	1	12	0,9700	0,2396	0,3108
	2	19	0,9708	0,2364	0,3068
	3	15	0,9667	0,2508	0,3273
	4	24	0,9708	0,2369	0,3069
ETH	1	18	0,6086	0,2106	0,2665
	2	15	0,5999	0,2123	0,2687
	3	16	0,5566	0,2212	0,2790
	4	17	0,6047	0,2118	0,2675

Tabela 7.3: Avaliação baseada no contexto das árvores-modelo do primeiro conjunto de experimentos com o M5P.

Ligante	Estratégia Pré-processamento	Avaliação		
		Precisão	*Recall*	*F-score*
NADH	1	0,1176	0,0385	0,0580
	2	0,4375	0,1346	0,2059
	3	0,3636	0,0769	0,1270
	4	0,1875	0,0576	0,0882
PIF	1	0,2143	0,1731	0,1915
	2	0,5294	0,3462	0,4186
	3	0,4667	0,1346	0,2090
	4	0,4571	0,3076	0,3678
TCL	1	0,1282	0,0962	0,1099
	2	0,4412	0,2885	0,3488
	3	0,4286	0,1154	0,1818
	4	0,3928	0,2115	0,2750
ETH	1	0,3939	0,2500	0,3059
	2	0,4375	0,2692	0,3333
	3	0,1250	0,0192	0,0333
	4	0,4516	0,2692	0,3373

todas as métricas, enquanto que para os demais ligantes, os melhores resultados alternam entre as estratégias 1 e 2. Por outro lado, as métricas baseadas no contexto (Tabela 7.3) mostram que os melhores resultados são para a estratégia 2 para os ligantes NADH, PIF e TCL [WIN11].

Para analisar esses resultados em termos de suas significância estatística, foi aplicado o teste de Friedman com um nível de significância de $\alpha = 0,05$ em ambas as Tabelas. Foram avaliados os valores MAE e RMSE da Tabela 7.2 e o F-score da Tabela 7.3. Na Tabela 7.3 foi avaliada se a estratégia 3 (CFS) era significativamente pior que as demais. Foram obtidos os valores de $p = 0,04$ para o MAE e $p = 0,054$ para o RMSE, o que indica que provavelmente essa estratégia é realmente pior que as estratégias 1, 2 e 4. Em relação ao F-score da Tabela 7.3, foi avaliado se a estratégia 2 é significativamente melhor que as outras. Neste caso, podemos afirmar que é verdade, já que foi obtido o valor de $p = 0,014$. Assim, é possível inferir que a abordagem de seleção de atributos baseada no contexto melhora os resultados iniciais. E, como o interesse nessa pesquisa são os modelos induzidos, a métrica de avaliação também baseada no contexto se mostra como mais adequada [WIN11].

7.3 Segundo Conjunto de Experimentos Utilizando o Algoritmo M5P

O segundo conjunto de experimentos utilizando o M5P tem por objetivo efetivamente selecionar conformações mais promissoras do modelo FFR do receptor. Para isso, foi gerado um novo arquivo de entrada, que seguiu a estratégia de seleção de atributos baseada no contexto, indicada pelos resultados do primeiro conjunto de experimentos com o M5P como a melhor estratégia de seleção de atributos, mas utilizou uma margem maior de distância entre resíduos do receptor e ligante, de 5,0 $\mathrm{\AA}$ para abranger um número maior de resíduos do receptor. Assim, o total de atributos de distância considerados para cada ligante foi de 106, 122, 121 e 128 para o NADH, PIF, TCL e ETH respectivamente.

O algoritmo M5P foi então aplicado para os 4 arquivos de entrada utilizando o parâmetro *Minimum number of instances* com o valor 1.000 e os demais parâmetros com valores *default*. Para exemplificar o resultado do segundo conjunto de experimentos com o M5P é apresentada na Figura 7.2 a árvore modelo obtida para o ligante NADH, que contém 10 nodos e 11 LMs. Essa Figura apresenta a árvore no formato do arquivo de saída do algoritmo M5P do WEKA. As demais árvores-modelo estão descritas no Apêndice B.

Os resultados das métricas preditivas para os 4 ligantes no segundo conjunto de experimentos com o M5P estão descritos na Tabela 7.4. A descrição dessa tabela é mesma da Tabela 7.2.

Tabela 7.4: Avaliação preditiva das árvores-modelo do segundo conjunto de experimentos com o M5P.

Ligante	Atributos	Nodos	Correlação	MAE	RMSE
NADH	107	10	0,9510	1,0198	1,4027
PIF	123	18	0,9689	0,3048	0,404
TCL	122	22	0,9707	0,2369	0,3072
ETH	129	18	0,6022	0,2124	0,2681

Em relação aos valores de correlação foram gerados modelos bastante satisfatórios. Esses modelos geraram árvores pequenas e por esse motivo possíveis de serem interpretadas. Para o NADH,

```
THR100 <= 11.49 :
|   HIE92 <= 8.549 :
|   |   ALA259 <= 9.563 :
|   |   |   TRP229 <= 7.255 : LM1 (864/37.388%)
|   |   |   TRP229 >  7.255 : LM2 (541/32.232%)
|   |   ALA259 >  9.563 :
|   |   |   LEU62 <= 2.916 : LM3 (1309/20.751%)
|   |   |   LEU62 >  2.916 :
|   |   |   |   ILE214 <= 10.617 :
|   |   |   |   |   ILE46 <= 3.535 : LM4 (359/30.83%)
|   |   |   |   |   ILE46 >  3.535 :
|   |   |   |   |   |   VAL188 <= 9.005 : LM5 (568/27.409%)
|   |   |   |   |   |   VAL188 >  9.005 : LM6 (662/44.15%)
|   |   |   |   ILE214 >  10.617 :
|   |   |   |   |   THR38 <= 6.779 :
|   |   |   |   |   |   ASP63 <= 5.247 : LM7 (732/24.057%)
|   |   |   |   |   |   ASP63 >  5.247 : LM8 (894/34.579%)
|   |   |   |   |   THR38 >  6.779 : LM9 (991/33.991%)
|   HIE92 >  8.549 : LM10 (1879/31.253%)
THR100 >  11.49 : LM11 (2485/26.418%)
```

Figura 7.2: Árvore modelo do ligante NADH para o experimento 2.

TCL e PIF foram obtidos mais de 95 % de correlação. O pior valor de correlação foi para o ligante ETH, em torno de 60 %, que pode ser explicada por este ligante ser uma pró-droga, ele se liga ao receptor InhA como um aduto com o NADH (ETH-NADH), e nestes experimentos ele foi considerado sem o NADH. Dessa forma, o ETH explora uma região do sítio de ligação do receptor que na verdade não está acessível pela presença do NADH. Não considerando essa diferença de correlação se comparada com os demais ligantes, 60 % é um valor de correlação satisfatório para validar o modelo obtido.

7.3.1 Resultados Obtidos - Análise das Árvores Modelo

As métricas preditivas consideradas mostram a qualidade dos modelos obtidos. Entretanto, como o objetivo dos experimentos utilizando o M5P era de selecionar conformações do receptor, foi preciso estabelecer um critério de seleção de LMs e uma metodologia de análise das árvores modelo obtidas. Essa metodologia consiste em identificar quais são os melhores LMs de cada modelo, para então percorrer as árvores e selecionar as conformações que pertencem os LMs selecionados. As instâncias classificadas nas LMs selecionadas indicam as conformações do receptor mais promissoras para serem utilizadas em docagem molecular com outros ligantes. Os 3 principais passos da metodologia desenvolvida são:

1. percorre-se as árvores modelo utilizando-se o conjunto de teste para identificar que instâncias pertencem a cada um dos LMs;

2. cuidadosamente define-se um critério de seleção dos LMs mais representativos;

3. avalia-se se as conformações selecionadas são de fato promissoras.

O conjunto de teste considerado é o de instâncias *BEST FEB*, ou seja, das instâncias com somente o melhor valor de FEB de cada simulação de docagem. Esse conjunto foi escolhido pois

cada conformação do receptor está relacionada com somente uma instância no conjunto de teste. Cada ligante tem seu conjunto de teste.

Sendo assim, iniciou-se pela implementação de *scripts Python* que mapeassem as instâncias dos conjuntos de teste para os respectivos LMs das árvores modelo de cada ligante (Figura 7.3(a)). Esses *scripts* (um para cada árvore modelo), ao serem executados, verificam a qual LM cada uma das instâncias do conjunto de teste pertence, gerando uma lista que relaciona conformação com LM (Figura 7.3(b)). A seguir, as conformações que pertencem ao mesmo LM são agrupadas e a média de FEB de cada grupo é calculada (Figura 7.3(c)). Por fim, tem-se condições de indicar quais LMs são mais representativos, e então utilizá-las para a seleção de conformações. Os LMs mais representativos são aquelas cuja média de FEB das instâncias do LM é menor do que a média de todas as instâncias do conjunto de teste (Figura 7.3(d)). Decidiu-se por esse critério de seleção pois, se a média de FEB do grupo relacionado com determinado LM é menor do que a média do todo é porque agrupou instâncias com valores de FEB bem negativos, ou seja, justamente os que queremos selecionar. Um exemplo desse processo está na Figura 7.3 para o conjunto de teste e árvore modelo do ligante NADH que ao final do processo somente o LM11 é selecionado.

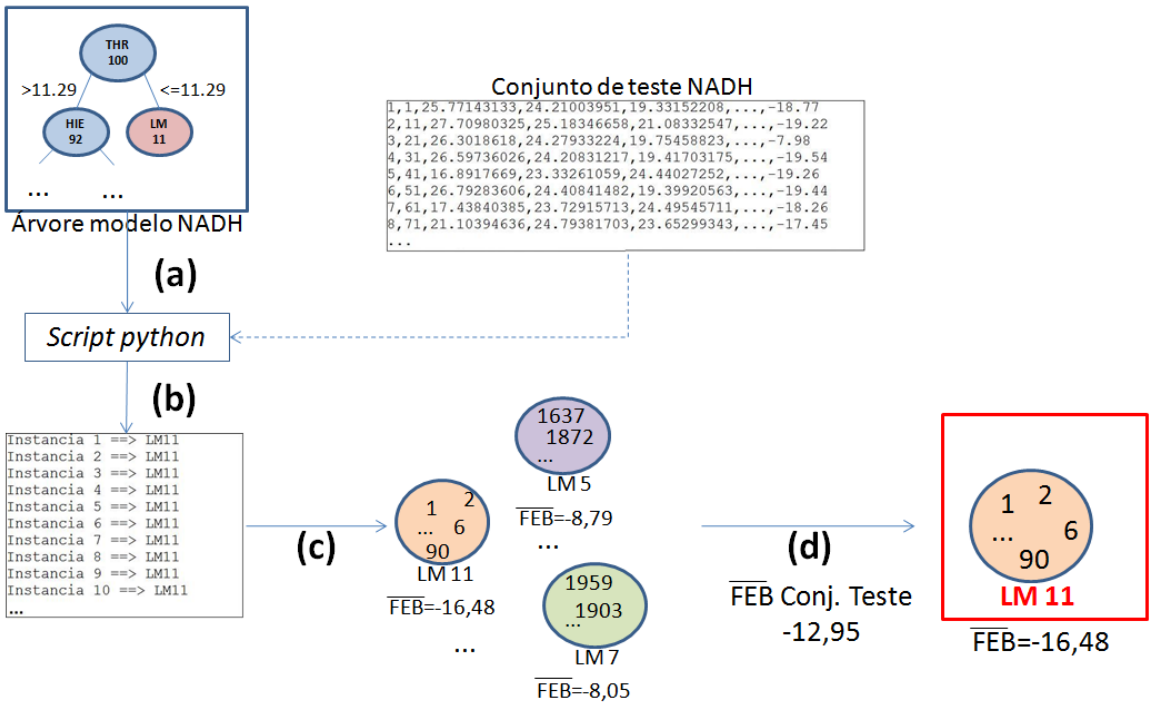


Figura 7.3: Representação esquemática da metologia utilizada para a seleção de LMs representativos.

Para exemplificar a metodologia de seleção de LMs considerou-se os resultados para o ligante PIF. A Tabela 7.5 apresenta esses resultados onde as colunas 1 e 4 contém os LMs, as colunas 2 e 5 o total de instâncias em cada LM e nas colunas 3 e 6, a média de FEB das instâncias de cada LM. Os LMs selecionados estão destacados na tabela.

Baseado na metodologia proposta, a média de FEB para o PIF no conjunto de teste é de -9,9 Kcal/mol (Tabela 3.1), os LMs selecionados para esse ligante são LM1, LM2, LM3, LM5 e LM7. Os resultados dos LMs selecionados para cada um dos demais ligantes estão descritos nas Tabelas

7.6, 7.7 e 7.8 para o NADH, TCL e ETH respectivamente, cuja descrição é a mesma da Tabela 7.5. Por exemplo, para o NADH, a média de FEB é de -12,9 kcal/mol, o que seleciona somente o LM11 como promissora.

Tabela 7.5: Análise dos LMs gerados para o ligante PIF.

LM	Total Instâncias	Média FEB Kcal/mol	LM	Total Instâncias	Média FEB Kcal/mol
LM1	1,776	-9,98	LM11	250	-9,65
LM2	91	-10,28	LM12	131	-9,57
LM3	48	-10,15	LM13	26	-9,76
LM4	96	-9,74	LM14	14	-9,32
LM5	65	-9,93	LM15	3	-8,98
LM6	178	-9,79	LM16	11	-4,88
LM7	105	-9,90	LM17	6	-4,78
LM8	38	-9,77	LM18	0	-
LM9	60	-9,71	LM19	2	-4,44
LM10	142	-9,53			

Tabela 7.6: Análise dos LMs geradas para o ligante NADH.

LM	Total Instâncias	Média FEB Kcal/mol	LM	Total Instâncias	Média FEB Kcal/mol
LM1	257	-10,67	LM7	53	-8,06
LM2	153	-8,43	LM8	141	-7,71
LM3	255	-9,39	LM9	87	-6,84
LM4	101	-9,82	LM10	66	-5,86
LM5	105	-8,79	LM11	1.521	-16,48
LM6	84	-7,82			

Tabela 7.7: Análise dos LMs geradas para o ligante TCL.

LM	Total Instâncias	Média FEB Kcal/mol	LM	Total Instâncias	Média FEB Kcal/mol
LM1	522	-9,03	LM13	27	-8,63
LM2	49	-8,94	LM14	30	-8,45
LM3	145	-8,97	LM15	17	-8,53
LM4	24	-8,81	LM16	78	-8,66
LM5	927	-8,90	LM17	88	-9,08
LM6	162	-8,84	LM18	315	-8,86
LM7	34	-8,76	LM19	49	-8,89
LM8	29	-8,72	LM20	107	-8,71
LM9	44	-8,64	LM21	27	-8,78
LM10	58	-8,82	LM22	49	-8,54
LM11	37	-8,52	LM23	2	-4,96
LM12	17	-8,68			

Tabela 7.8: Análise dos LMs geradas para o ligante ETH.

LM	Total Instâncias	Média FEB Kcal/mol	LM	Total Instâncias	Média FEB Kcal/mol
LM1	1,263	-6,71	LM11	6	-6,18
LM2	517	-6,62	LM12	17	-6,39
LM3	48	-6,65	LM13	321	-7,18
LM4	47	-6,52	LM14	243	-7,03
LM5	12	-6,47	LM15	43	-6,97
LM6	6	-6,26	LM16	137	-7,01
LM7	5	-6,21	LM17	137	-6,93
LM8	14	-6,48	LM18	21	-6,80
LM9	2	-6,35	LM19	177	-6,75
LM10	27	-6,56			

Para verificar se as conformações selecionadas são realmente as conformações com melhores resultados de docagem molecular, os seus valores de FEB foram cuidadosamente avaliados. Para isso, as instâncias dos conjuntos de teste de cada ligante foram organizadas em uma ordem ascendente por FEB. Então, compararam-se as conformações no topo da lista ordenada (ou seja, os de FEB mais negativa) com as conformações dos LMs selecionadas. Os resultados obtidos estão descritos na Tabela 7.9. Na coluna 1 tem-se os ligantes, nas colunas 2, 3 e 4 o número total de conformações selecionadas que estão no TOP 10, 100 e 1000 da lista ordenada por FEB, respectivamente. A coluna 5 mostra o total de conformações selecionadas para cada ligante.

Tabela 7.9: Resultados das análises dos LMs selecionadas e suas conformações para os 4 ligantes.

Ligante	Top 10 lista FEB	Top 100 lista FEB	Top 1000 lista FEB	Total de conformações selecionadas/ Total de conformações
NADH	10	100	998	1.521 / 2.823
TCL	10	100	610	1.780 / 2.837
PIF	10	100	1.000	2.085 / 3.042
ETH	10	92	617	902 / 3.043

Baseado nos dados descritos na Tabela 7.9 nota-se que as conformações selecionadas são conformações que obtiveram bons resultados em docagem molecular para os 4 ligantes. Para o NADH e PIF, dos 10, 100 e 1.000 TOP melhor FEB, a metodologia proposta selecionou quase 100% das melhores conformações. Para a ETH, foram selecionados os 10 melhores, 92 % dos 100 melhores e 617 dos 1.000 melhores, porém esse ligante foi o que selecionou menos conformações. Os piores resultados foram para o TCL, onde das 1.780 conformações selecionadas, somente 610 estão entre as 1000 melhores para esse ligante.

Dessa forma, nesse segundo conjunto de experimentos com o algoritmo M5P a maior contribuição é a estratégia de seleção de conformações que foi capaz de selecionar as conformações mais promissoras. Além disso, as análises das árvores modelo indicam quais são os resíduos do receptor mais importantes para a determinação de bons e ruins valores de FEB. Por exemplo, a partir da

árvore do NADH (Figura 7.1) e do LM selecionado para esse ligante (Tabela 7.6) é possível observar que todas as conformações do receptor cuja distância do resíduo THR100 é maior do que 11,49 $\mathring{A}$ são consideradas conformações promissoras. A discussão de como essa informação será utilizada para a busca de novos inibidores para essa enzima está fora do escopo desse trabalho e consiste em trabalhos futuros a serem realizados.

7.4 Considerações Finais

Neste trabalho, a partir dos arquivos de entrada descritos no Capítulo 5, que contém os 268 atributos de distâncias mínimas entre os 268 resíduos do receptor e cada um dos ligantes e o atributo-alvo FEB, foram gerados novos arquivos de entrada a partir de diferentes estratégias de seleção de atributos. A primeira estratégia de seleção de atributos é proposta em [WIN10c, WIN11] e é baseada no contexto. A segunda estratégia utiliza o algoritmo de aprendizagem de máquina CFS (Correlation based Feature Selection) para a seleção automática de atributos. E a terceira estratégia de seleção combina as duas primeiras. São então comparados os resultados do algoritmo M5P para as 3 diferentes entradas utilizando as métricas clássicas RMSE, MAE e Correlação assim como utilizando métricas também baseadas no contexto. Analisando estatisticamente os resultados obtidos com o algoritmo M5P com o Teste de Friedman, observou-se que a abordagem baseada no contexto melhorou significativamente as métricas dos modelos gerados a partir das diferentes entradas, enquanto que a seleção de atributos com o algoritmo CFS obteve os piores resultados em relação as métricas de avaliação preditivas. Dessa forma, os resultados do primeiro conjunto de experimentos com o M5P mostram o quanto é importante o pré-processamento para a obtenção de modelos mais acurados e interpretáveis. Como trabalho futuro, pretende-se utilizar as informações dos melhores modelos gerados para a seleção de novos compostos candidatos baseado em como o modelo FFR interage com os ligantes já estudados.

Para o segundo conjunto de experimentos com o M5P foram utilizados arquivos de entrada com seleção de atributos baseada no contexto mas com uma distância de 5,0 $\mathring{A}$ do ligante. Os resultados do segundo conjunto de experimentos com o M5P foram modelos que de acordo com as métricas de avaliação preditivas são bons modelos. Com estes modelos, foi aplicado um pós-processamento nas árvores modelo geradas onde, para cada LM foi calculado a média de FEB das instâncias associadas a esse LM. A partir desses valores foi determinado que um LM é representativo se a média de FEB é menor ou igual a média de FEB do conjunto de teste. As instâncias nos LMs selecionados são então consideradas como mais promissoras, o que totalizou 1.521 conformações para o NAD, 1.780 para o TCL, 2.085 para o PIF e 902 para o ETH. A metodologia de pós-processamento apresentada permitiu o desenvolvimento de um critério de seleção de LMs, que por sua vez, foram capazes de selecionar esse conjunto de conformações do receptor mais promissoras.

Sendo assim, as maiores contribuições desse capítulo dizem respeito ao pré-processamento e avaliação dos modelos baseados no contexto, que produziram melhores modelos e a metodologia de pós-processamento que permitiu a indicação de conformações do receptor mais promissoras para cada

um dos ligantes. Apesar dos resultados de todos os experimentos com o M5P serem interessantes, a utilização dos mesmos, diretamente para seleção de conformações em futuros experimentos de docagem molecular não é promissora. O maior problema encontrado foi de que as conformações mais promissoras eram diferentes para cada um dos ligantes, o que dificulta a utilização das mesmas para análises de interação com novos compostos obtidos de bancos de compostos como o ZINC [IRW05]. Ou seja, não é possível, a partir desses resultados, estabelecer um conjunto único de conformações mais relevantes. Outro problema encontrado é que, para se utilizar os modelos gerados para predizer o valor de FEB de novos ligantes é necessário saber as distâncias mínimas dos mesmos para os resíduos do receptor, informação que somente é obtida após a execução da docagem molecular, o que também dificulta a utilização dos modelos gerados com o M5P para efetivamente selecionar conformações do receptor para compostos ainda não testados.

Pelos motivos descritos acima optou-se por não mais se utilizar como entrada nos experimentos com mineração de dados, os resultados de docagem molecular e sim, diretamente, as conformações do receptor FFR. E, como não será mais utilizado os resultados de docagem, não tem-se mais um atributo-classe FEB. Assim, a técnica de mineração a ser aplicada deverá ser de aprendizado não-supervisionado, na qual a classe de cada instância é desconhecida assim como o total de grupos e a estrutura dos mesmos. A técnica de aprendizado não-supervisionado escolhida é a de Agrupamento. Para a utilização dessa técnica de mineração de dados não foi utilizado os algoritmos de agrupamento implementados no WEKA e sim os algoritmos de agrupamento descritos em [SHA07], que estão implementados no módulo Ptraj9. Os experimentos de agrupamento serão descritos no próximo capítulo.

8. RESULTADOS 4 - APLICAÇÃO DE ALGORITMOS DE AGRUPAMENTO

Este capítulo descreve os experimentos realizados com a técnica de mineração de dados não-supervisionada Agrupamento. O principal objetivo desse conjunto de experimentos é de agrupar conformações mais similares do modelo FFR do receptor, onde a entrada dos algoritmos de agrupamento são as próprias conformações. Os agrupamentos gerados nesses experimentos são então utilizados pelo padrão P-MIA, proposto em [HÜB10] para ser utilizado em Bioinformática com o propósito de reduzir a quantidade total de conformações a serem processadas em experimentos de docagem molecular com o modelo FFR e garantindo que as melhores conformações continuariam a ser consideradas. Assim, a partir dos grupos de conformações obtidos, após a execução do agrupamento, e utilizando o P-MIA [HÜB10], há um ganho em relação a quantidade de conformações que não precisam ser processadas, o que é identificado dinamicamente, sem interferência do usuário e realizado em paralelo, aumentando o desempenho desse tipo de experimento e permitindo que novos compostos sejam testados com um tempo de processamento reduzido.

A técnica de agrupamento já foi utilizada em vários trabalhos para o agrupamento de conformações do receptor resultantes de DM. Um exemplo é o trabalho apresentado por Torda e van Gunsteren [TOR94] onde 2 algoritmos clássicos de agrupamento *Single Linkage* e *Hierarquical* são aplicados a um subconjunto de átomos que representam as conformações de uma trajetória de simulação pela DM. Mais recentemente, Shao *et al.* [SHA07] implementaram 11 algoritmos de agrupamento (*Average Linkage, Bayesian, Centripetal, Centripetal Complete, COBWEB, Complete Linkage, Edge Linkage, Hierarchical, K-means, Linkage* e *SOM*) onde comparam seus resultados e os utilizam para entender os dados de simulações pela DM.

Os algoritmos de agrupamento utilizam diferentes funções de similaridade para determinar a proximidade dos dados do conjunto de entrada. O tipo de função de similaridade deve estar de acordo com os dados de entrada [HAN06]. Nos trabalhos [TOR94] e [SHA07], a medida de similaridade utilizada por todos os algoritmos de agrupamento foi a de RMS das coordenadas cartesianas dos átomos considerados. Essa medida, definida por $D_{ab}RMS$, corresponde a soma dos quadrados das distâncias sobre todos os pares ij de N átomos que estão sendo considerados das conformações a e b (d_{ij} é a distância tridimensional entre os átomos i e j):

$$D_{a,b}RMS = \sqrt{\frac{2}{N(N-1)} \sum_{i<j}^{N} (d_{ij}^{a} - d_{ij}^{b})^2} \tag{8.1}$$

Além da execução de experimentos de agrupamento visando a redução do custo computacional de docagem com o FFR, nesta Tese propõe-se a definição de novas funções de similaridade desenvolvidas com o objetivo de agrupar as conformações do receptor de forma mais eficaz. Essas novas funções de similaridade são definidas neste capítulo juntamente com a descrição de todos

os experimentos de agrupamento executados. Esses foram realizados com diferentes conjuntos de átomos de entrada uma vez que, de acordo com Shao et al. [SHA07], os resultados dos algoritmos de agrupamento são fortemente dependentes da escolha de átomos para a comparação par-a-par da função de similaridade.

Esse capítulo compreende: os testes realizados com os algoritmos implementados por [SHA07] para a determinação do número de grupos; a descrição de como os dados de saída do programa LigPlot foram preparados para serem utilizados nas novas funções de similaridade e como essas foram definidas. Após são descritos todas as configurações de experimentos de agrupamento executados, onde os resultados obtidos com a função clássica RMS são comparadas com os das funções desenvolvidas. Por fim, são apresentadas análises utilizando o P-MIA [HÜB10] que comparam a função RMS com uma das funções desenvolvidas mostrando efetivamente o ganho de processamento obtido com a utilização do P-MIA em conjunto com os resultados dos algoritmos de Agrupamento.

8.1 Determinação do Número de Grupos

O primeiro conjunto de experimentos de Agrupamento aplica os 10 diferente algoritmos implementados em [SHA07] com a função de similaridade RMS definida por $D_{ab}RMS$ (Equação 8.1). Estes foram executados com o objetivo de melhorar o entendimento a respeito da implementação destes algoritmos assim como, para estabelecer o total de grupos ideal.

Os 10 algoritmos utilizados foram: *Average Linkage (Average), Bayesian, Centripetal, Centripetal Complete (Centripetal_Comp), Complete Linkage (Complete), Edge Linkage (Edge), Hierarchical, Linkage, K-means* e *SOM*. Em todos os experimentos de agrupamento desta Tese a DM utilizada como entrada é a de 3.100 ps do receptor InhA descrita na Seção 3.4 do Capítulo 3. Nestes primeiros experimentos, os 10 algoritmos foram executados com 2 conjuntos de átomos de entrada:

1. ALL - Considera os átomos de Carbono-α dos 268 resíduos do receptor. Ou seja, nos experimentos de agrupamento com essa entrada, a função de similaridade entre duas conformações é calculada como a soma sobre todos os pares ij dos $N = 268$ átomos;

2. 25_RES - Considera os átomos de Carbono-α dos 25 resíduos selecionados na análise descrita no Capítulo 5 como os que mais interagem com os ligantes estudados. Nesse caso, a função é calculada sobre $N = 25$ átomos de cada conformação considerada.

8.1.1 Testes com 10-100 Agrupamentos

Esse estudo inicial tinha por objetivo analisar se as entradas ALL e 25_RES para os algoritmos causavam diferentes resultados, principalmente em relação as métricas de avaliação de Agrupamento DBI e pSF descritas no Capítulo 4, Seção 4.3.3.11. A Figura 8.1 mostra os gráficos da métrica DBI e a Figura 8.2 contém os mesmos gráficos para a métrica pSF.

Nas Figuras 8.1 e 8.2 têm-se os resultados para experimentos de agrupamento com 10, 20, 30, 40, 50, 60, 70, 80 90 e 100 grupos, para os 10 algoritmos. Esses resultados mostram que:

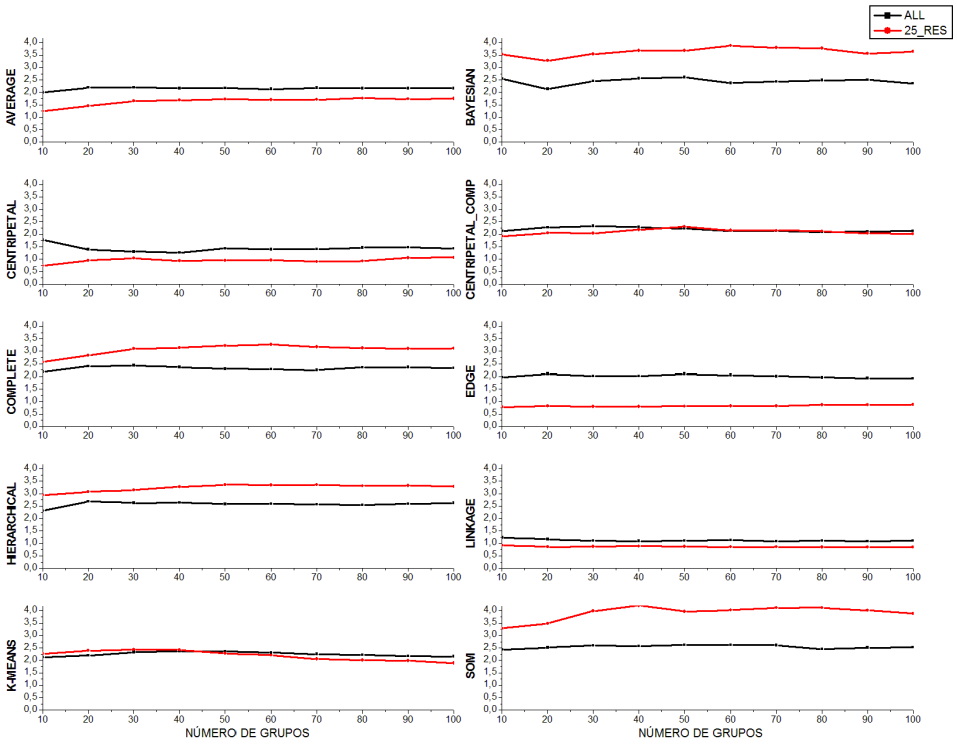


Figura 8.1: Gráficos da métrica DBI dos agrupamentos para os 10 algoritmos considerando 10, 20, 30, 40, 50, 60, 70, 80, 90 e 100 grupos. Em preto os resultados para entrada ALL e em vermelho para a entrada 25_RES.

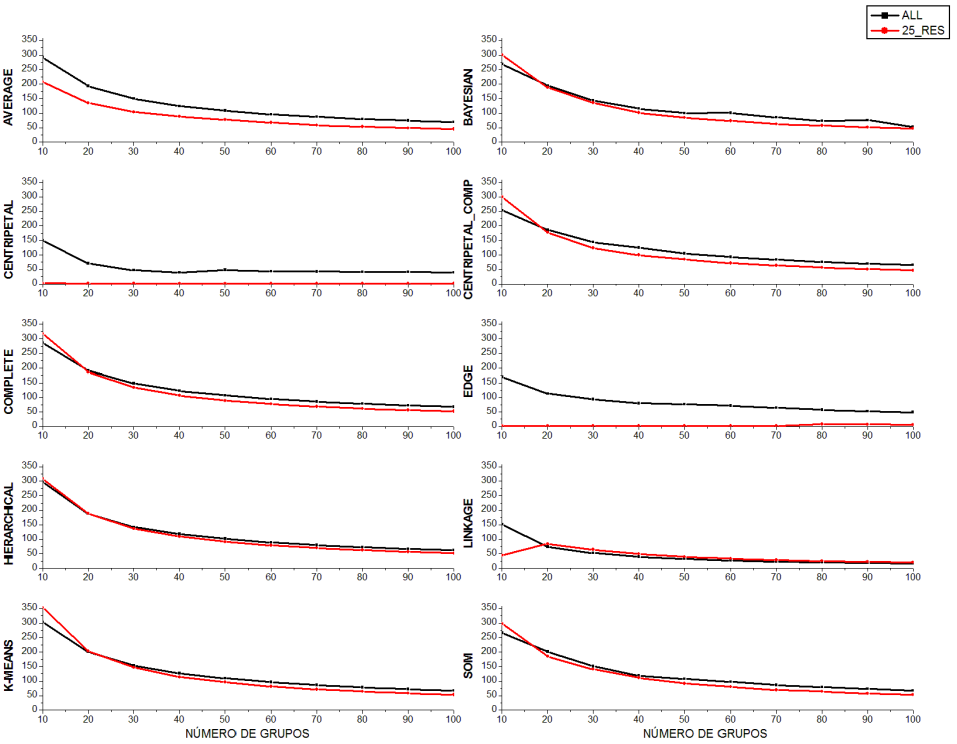


Figura 8.2: Gráficos da métrica pSF dos agrupamentos para os 10 algoritmos considerando 10, 20, 30, 40, 50, 60, 70, 80, 90 e 100 grupos. Em preto os resultados para entrada ALL e em vermelho para a entrada 25_RES.

- A partir de aproximadamente 30 grupos não há mais modificação nos valores de ambas as métricas.

- Em relação a métrica pSF não há muita diferença entre os valores para os conjuntos de átomos de entrada ALL e 25_RES, com exceção para os algoritmos *Centripetal*, *Linkage* e *Edge* que apresentam valores próximos de zero para a 25_RES. Para a métrica DBI, as entradas diferem mais para os 10 algoritmos, porém alternam em melhores resultados. Logo, não é possível afirmar o melhor conjunto de átomos de entrada para os experimentos de agrupamento. Assim, ambas as entradas serão utilizadas nos experimentos subsequentes.

- Nesses resultados, os melhores resultados em relação a métrica DBI foram obtidos para os algoritmos *Centripetal*, *Linkage* e *Edge*. Entretanto, os 3 algoritmos apresentam valores de pSF ruins, principalmente para a entrada 25_RES. Dessa forma, foi decidido continuarmos utilizando os 10 algoritmos nos demais experimentos, pois somente com este primeiro conjunto de resultados não há como afirmar quais algoritmos apresentam melhor comportamento para os diferentes números de grupos e entradas.

- Para a métrica DBI, a partir de 20 grupos, os valores tendem a se manter iguais. Em relação a pSF, os melhores valores, que correspondem aos maiores valores de pSF se mantém até no máximo 30 grupos, sendo melhores para até 20 grupos. Assim, foi estabelecido que para mais de 20 grupos não há mais ganho nos agrupamentos para ambas as entradas em todos os algoritmos.

8.1.2 Testes com 2-20 Agrupamentos

Para um entendimento mais detalhado dos resultados, para agrupamentos de até 20 grupos, foi realizado o segundo conjunto de experimentos com os mesmos conjuntos de átomos de entrada ALL e 25_RES, mas com 2 até 20 grupos, variando de 1 em 1. Ou seja, foram realizados testes para os 2 conjuntos de átomos de entrada, para os 10 algoritmos, considerando 2, 3, 4, ... até 20 grupos. Os resultados estão descritos nos gráficos das Figuras 8.3 e 8.4 para as métricas DBI e pSF respectivamente. Em preto nos gráficos têm-se os resultados para entrada ALL e em vermelho os resultados para a entrada 25_RES.

Como se pode perceber analisando os resultados dos gráficos, os valores de DBI foram em média menores, atingindo um máximo de 3,5 (para os experimentos 10-100 o máximo foi de 4,0) e os valores de pSF foram bem maiores, atingindo valores máximos de aproximadamente 1.000 quando o máximo com os experimentos de 10-100 grupos foi de 300 para essa métrica. Essa análise indica que os resultados com até, no máximo, 20 grupos apresentam melhores valores de DIB e pSF do que a análise inicial com um mínimo de 10 grupos.

Assim como para o primeiro conjunto de experimentos, apesar de haver variação nos resultados para as 2 entradas ALL e 25_RES, não é possível estabelecer um consenso de qual conjunto de átomos de entrada gera melhores agrupamentos pois há alternância entre os melhores valores de

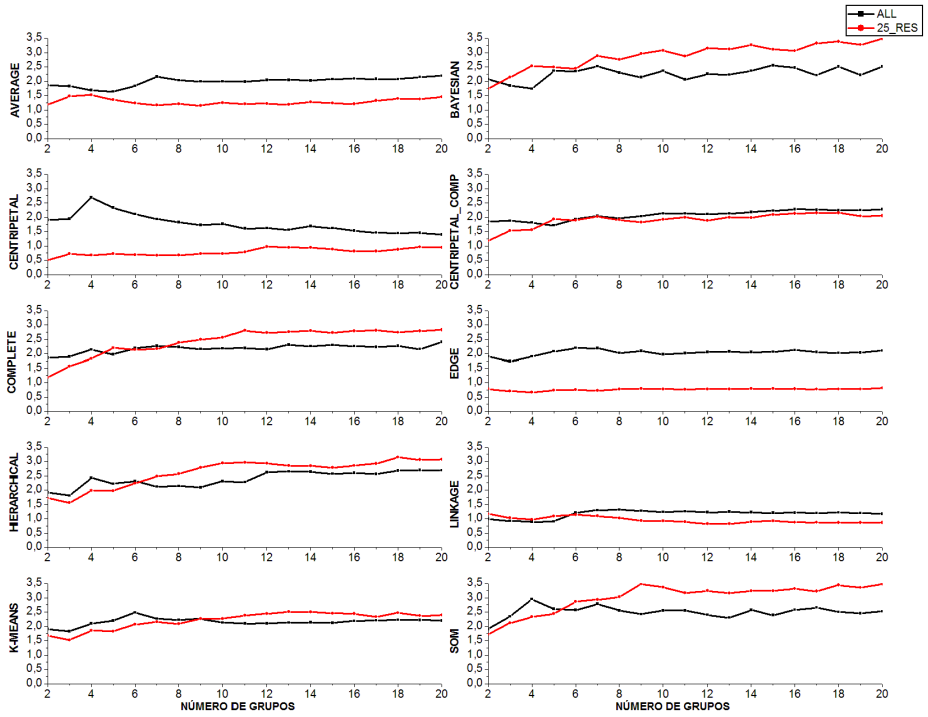


Figura 8.3: Gráficos da métrica DBI dos agrupamentos para os 10 algoritmos considerando 2-20 grupos, com os conjuntos de átomos de entrada ALL e 25_RES.

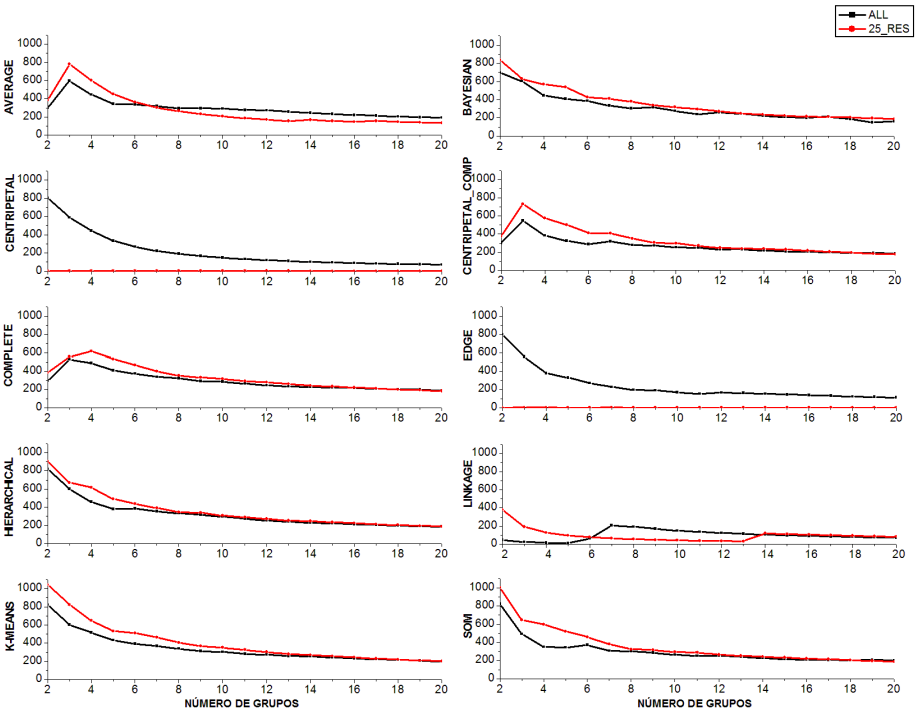


Figura 8.4: Gráficos da métrica pSFdos agrupamentos para os 10 algoritmos considerando 2-20 grupos, com os conjuntos de átomos de entrada ALL e 25_RES.

DBI e pSF entre ambas as entradas. Com esse estudo mais detalhado dos resultados para 2-20 grupos é possível concluir que com mais do que 10 grupos não se tem mais ganho em ambas as métricas, os valores de DBI se mantém iguais e os valores de pSF tendem somente a diminuir para

os 10 algoritmos. Dessa forma, a partir desses gráficos das Figuras 8.3 e 8.4 decidiu-se estabelecer o máximo de 10 grupos para os experimentos subsequentes.

Apesar dos algoritmos *Centripetal*, *Edge* e *Linkage* apresentarem valores ruins de pSF independente do total de grupos, isso somente ocorre para a entrada 25_RES. Como decidiu-se continuar a utilização de ambos conjuntos de átomos de entrada, esses algoritmos não foram descartados por apresentarem bons resultados para a entrada ALL. Assim, resumidamente, estabeleceu-se que nos experimentos seguintes de agrupamento:

- devem ser testados agrupamentos de 2 até no máximo 10 grupos;
- serão utilizadas ambas as entradas ALL e 25_RES;
- os 10 algoritmos devem ser considerados nos experimentos.

8.2 Funções de Similaridade

Para as novas funções de similaridade decidiu-se incorporar ao valor de distância $D_{a,b}RMS$ informações adicionais que permitissem uma melhora na determinação da similaridade entre duas conformações. Para isso, optou-se por utilizar os resultados de processamento do software LigPlot (Seção 3.1.4 do Capítulo 3). O LigPlot, ao receber como entrada um arquivo .PDB com o receptor e o ligante, fornece como saída uma lista de contatos hidrofóbicos e ligações de Hidrogênio estabelecidos entre o complexo receptor-ligante.

Portanto, a partir do resultado do LigPlot, tem-se uma lista de quantos contatos cada conformação estabeleceu com determinado ligante. Essa informação é incorporada ao valor de distância entre conformações para que a forma como estas estabelecem contatos com um mesmo ligante auxilie na determinação da similaridade entre as mesmas. As novas funções de similaridade, assim como a descrição de como foram preparados os resultados do LigPlot para serem utilizados em conjunto com o valor de $D_{a,b}RMS$ são descritas nas próximas seções.

8.2.1 Preparação da Entrada para as Funções de Similaridade

Para realizar essa análise sobre como cada conformação estabelece contatos, foi necessário escolher qual ligante seria mais apropriado. Foram escolhidas 2 entradas diferentes para o LigPlot: uma entrada que considera o substrato THT e outra que utiliza o substrato THT e o ligante NADH. O substrato THT foi escolhido pois ele encontra-se na mesma região do sítio de ligação onde possíveis inibidores da InhA se ligam. E a entrada com o THT+NADH foi utilizada pois a região ocupada por estes na InhA compreende a maior parte do sítio de ligação deste receptor. Isso significa que, analisando como o THT ou o THT+NADH interagem com cada uma das conformações, pode-se ter uma visão de como outros compostos podem estabelecer contatos com este receptor.

8.2.1.1 Entrada Utilizando InhA + THT

Como explicado na Seção 3.1.4 do Capítulo 3, como entrada para o LigPlot é fornecido um arquivo PDB que contém o receptor e o(s) ligante(s). Sendo assim, para analisar os contatos estabelecidos entre cada uma das 3.100 conformações da DM da InhA e o substrato THT é necessário preparar 3.100 PDBs que contenham as estruturas da DM juntamente com a estrutura do THT, estando este posicionado corretamente no sítio de ligação do receptor. Para isso foram utilizados os seguintes arquivos:

- a estrutura da enzima InhA com código PDB:1BVR obtida no *Protein Data Bank* [BER00]. Essa estrutura da InhA foi determinada com o NADH e o THT e apresenta 6 cadeias A, B, C, D, E e F. Para a preparação dos arquivos para o LigPlot foi utilizada somente a Cadeia C deste arquivo PDB;
- a primeira estrutura da DM da InhA, denominada conformação_1.DM.

Utilizando o software VMD de visualização de estrutura de moléculas biológicas [HUM96] executou-se os seguintes passos:

1. as duas estruturas PDB foram abertas no VMD: 1BVR_cadeiaC e conformação_1.DM;
2. a estrutura 1BVR_cadeiaC é sobreposta na conformação_1.DM. Assim, o substrato THT fica posicionado corretamente na conformação_1.DM e consequentemente nas demais conformações da DM. O arquivo PDB do THT posicionado é armazenado;

A Figura 8.5 mostra o resultado desse posicionamento. A estrutura da 1BVR está em magenta, a estrutura conformação_1.DM está em vermelho. Em verde está o THT posicionado.

8.2.1.2 Entrada Utilizando InhA + THT + NADH

Para a preparação da segunda entrada para o programa LigPlot utilizou-se os seguintes arquivos:

- a cadeia C da estrutura da InhA com o código PDB:1BVR;
- a estrutura da InhA como código PDB:1ENY. Essa estrutura contém o ligante NADH e foi utilizada como estrutura inicial na DM considerada nesta Tese;
- a primeira estrutura da DM - conformação_1.DM;

Utilizando o software VMD preparou-se esses arquivos da mesma forma que o InhA+THT, mas neste segundo a estrutura 1ENY também foi sobreposta na conformação_1.DM, posicionando o ligante NADH corretamente na conformação_1.DM e consequentemente nas demais da DM. Os arquivos do THT e NADH posicionados são armazenados. A Figura 8.6 mostra o resultado desse posicionamento onde a estrutura da 1BVR está em magenta, a estrutura conformação_1.DM está em vermelho e a estrutura da 1ENY está em azul. Em verde está o THT posicionado e em amarelo o NADH.

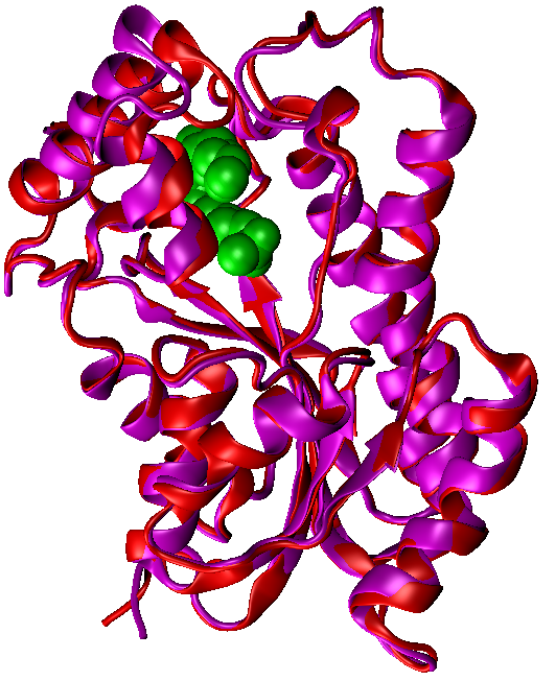


Figura 8.5: Posicionamento do substrato THT no sítio de ligação da estruturas conformação_1.DM. A estrutura da 1BVR está em magenta, a estrutura conformação_1.DM da DM está em vermelho, ambas na forma de *New_cartoon*. Em verde está o THT na forma de *VDW*.

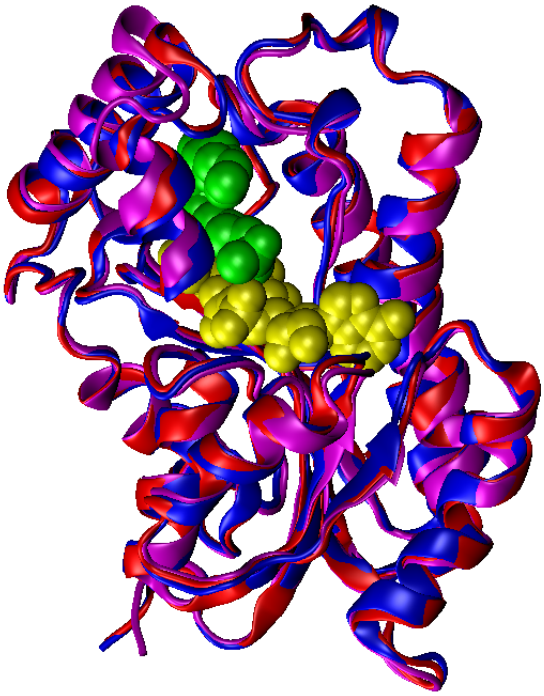


Figura 8.6: Posicionamento do substrato THT e do ligante NADH no sítio de ligação da estruturas conformação_1.DM. A estrutura da 1BVR está em magenta, a estrutura conformação_1.DM da DM está em vermelho, a estrutura da 1ENY está em azul, todas na forma de *New_cartoon*. Em verde está o ligante THT na forma de *VDW* e em amarelo o ligante NADH também na forma de *VDW*.

8.2.2 Execução do Programa LigPlot e Processamento de sua Saída

Para a execução do LigPlot com cada uma das 3.100 entradas foi desenvolvido um *script* que finaliza a preparação da entrada, executa o LigPlot e processa a saída. Esse *script* segue a estrutura do fluxograma descrito pela Figura 8.7 que utiliza como exemplo o THT.

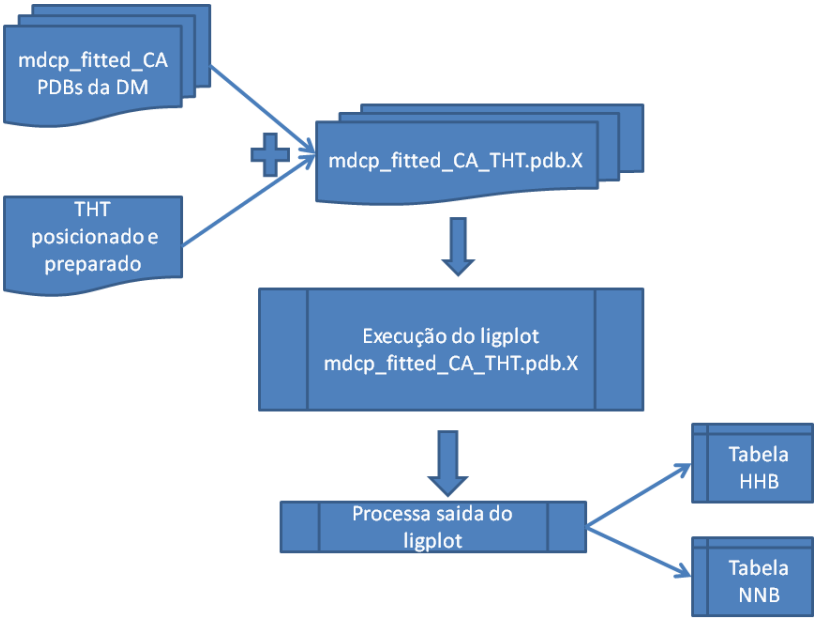


Figura 8.7: Fluxograma que descreve o script de execução do LigPlot.

1. para cada estrutura da DM, é concatenado no seu final o PDB do THT ou NAD+THT já posicionado. Esse PDB é então processado pelo LigPlot;

2. o resultado do LigPlot é um conjunto de arquivos, entre eles: LigPlot.hhb, LigPlot.nnb, LigPlot.sum, LigPlot.ps (Seção 3.1.4 do Capítulo 3). Destes arquivos são armazenados o LigPlot.sum e o LigPlot.ps. O LigPlot.sum é então processado para que os contatos estabelecidos entre cada conformação da DM e o THT ou THT+NADH sejam contabilizados em 2 tabelas de resultados: a Tabela HHB, que contém os totais de ligações de hidrogênio e a Tabela NNB com os totais de contatos hidrofóbicos.

Como exemplo das tabelas HHB e NNB é descrita a Tabela 8.1. Nessa tabela cada linha corresponde a uma conformação processada (InhA+THT ou InhA+THT+NADH) e cada coluna é um dos 268 resíduos do receptor. Em cada célula é armazenado o total de contatos que determinado resíduo do receptor estabeleceu com o ligante (THT ou NAD+THT). Na última coluna tem-se a soma dos contatos que determinada conformação estabeleceu com o ligante.

8.2.3 Funções Considerando o Total de Contatos Normalizado

As duas primeiras funções de similaridade desenvolvidas, chamadas de TCN e TCN_Mult2 utilizam os valores de totais de contatos estabelecidos por cada conformação com as 2 entradas

Tabela 8.1: Exemplo de tabela NNB resultante do processamento da saída LigPlot.sum do LigPlot.

Conformação	Res1	Res2	...	Res214	...	Res268	Total
1	0	0	...	5	...	0	66
2	0	0	...	4	...	0	74
...	...	...	...	...	...	...	...
3.100	0	0	...	6	...	0	55

analisadas: THT e THT+NAD. O objetivo das funções TCN e TCN_Mult2 é de modificar o valor de distância entre 2 conformações baseado em quantos contatos as mesmas estabeleceram.

8.2.3.1 Função TCN

A preparação dos dados resultantes do processamento do LigPlot (Tabela 8.1) para utilização na função TCN compreende as seguintes etapas (estes passos são executados para as 2 entradas separadamente):

1. As tabelas NNB e HHB foram analisadas para a determinação de quantos resíduos do receptor estabeleceram contato com o THT e com o THT+NADH. Todos os resíduos que não estabeleceram nenhum contato ao longo dos 3.100 resultados avaliados foram descartados, o que resultou nos resíduos descritos na Tabela 8.2. Os valores destacados nessa Tabela correspondem aos resíduos que são coincidentes com os Top 25 resíduos (análise descrita no Capítulo 5 dos resíduos que mais interagem com os ligantes estudados).

Tabela 8.2: Totais de resíduos do receptor que estabelecem contatos com o THT e THT+NADH baseado nos resultados do LigPlot.

Tabela LigPlot	Entrada	Resíduos	Total
HHB	THT	**THR195, ALA197**	2
NNB	THT	**MET97**, GLN99, MET102, **PHE148**, MET154, PRO155, ALA156, TYR157, **MET160**, PRO192 **THR195**, LEU196, **ALA197**, MET198, SER199 ALA200, ILE201, VAL202, LEU206, ILE214	20
HHB	THT+NADH	ILE14, **ILE15**, THR16, **SER19**, **ILE20** ALA21, LEU62, **MET97**, **MET146**, **LYS164** **ILE193, THR195**, LEU196, **ALA197**, MET198	15
NNB	THT+NADH	**GLY13**, ILE14, **ILE15**, THR16, SER18, **SER19** **ILE20, ALA21, PHE40, ARG42,** LEU62 ASP63, VAL64, **SER93, ILE94, GLY95** **PHE96, MET97**, GLN99, MET102, **ILE121** **SER122, MET146, ASP147, PHE148**, MET154 PRO155, ALA156, TYR157, **MET160, LYS164** **ALA190, GLY191, PRO192, ILE193,** ARG194 **THR195,** LEU196, **ALA197**, MET198, SER199 ALA200, ILE201, VAL202, LEU206, ILE214	46

2. Foram criadas 2 tabelas NNB+HHB, uma para cada uma das entradas THT e THT+NADH, que correspondem a soma dos contatos da tabela NNB com os contatos da tabela HHB.

3. Para cada uma das tabelas NNB+HHB foi calculado o total de contatos estabelecidos por cada conformação, somando-se os contatos de todos os resíduos. A partir dos valores dessa coluna com totais de contato, foi calculado o valor máximo de contatos (no exemplo para a entrada THT esse valor foi de 146 contatos). Com esse valor máximo, os demais valores de totais de contatos foram normalizados, conforme mostra a Tabela 8.3 de exemplo, que é resultado do processamento da entrada THT. Nessa tabela têm-se nas linhas as conformações do modelo FFR e nas colunas os resíduos do receptor, sendo a penúltima coluna, o totais de contatos de cada conformação e a última coluna corresponde ao total de contatos normalizado (V_TCN), baseado no valor máximo de contatos, que neste exemplo é o valor de 146.

Tabela 8.3: Parte da tabela de totais de contatos normalizados para a entrada THT+NADH. O valor de contatos máximo é 146.

Conf.	MET97	GLN99	MET102	...	ILE201	VAL202	LEU206	ILE214	Total	V_TCN
1	0	0	4	...	14	0	2	6	71	0,49
2	0	0	1	...	12	0	2	6	38	0,26
3	0	0	1	...	14	0	1	5	38	0,26
4	0	0	1	...	13	0	1	5	71	0,49
...	...	...	...	...	...	...	...	...	...	...
3.099	0	0	6	...	4	0	0	3	65	0,44
3.100	2	0	10	...	4	0	0	2	67	0,46

Para a função TCN foram utilizados os valores da primeira coluna (número da conformação) e da última coluna (totais de contato normalizados). Sendo a função RMS determinada por $D_{ab}RMS$ (Equação 8.1), a TCN utiliza como função de similaridade o valor de $D_{ab}TCN$, que consiste em:

$$D_{a,b}TCN = \frac{D_{a,b}RMS}{V\_TCN_a + V\_TCN_b} \tag{8.2}$$

Onde, V_TCN_a é o total de contatos normalizado da conformação a e V_TCN_b é o valor total de contatos normalizados da conformação b.

Essa função foi pensada dessa forma tentando agrupar conformações que tenham mais contatos, diminuindo a distância entre ambas e aumentando a distância entre conformações que apresentem ambas poucos contatos. Por exemplo, sendo conformação_1, conformação_2 e conformação_4 três conformações que determinado algoritmo de agrupamento está verificando a similaridade. Supondo que o valor de $D_{1,2}RMS$ e $D_{1,4}RMS$ entre as conformações seja de 1,0 Å. Sendo os valores de V_TCN das conformações os descritos na Tabela 8.3, tem-se:

$$D_{1,2}TCN = \frac{D_{1,2}RMS}{V\_TCN_1 + V\_TCN_2} = \frac{1}{0,49 + 0,26} = 1,33 \tag{8.3}$$

$$D_{1,4}TCN = \frac{D_{1,4}RMS}{V\_TCN_1 + V\_TCN_4} = \frac{1}{0,49 + 0,49} = 1,02 \quad (8.4)$$

Com este exemplo, é possível ver que, como a conformação_1 tem um V_TCN igual a $0,49$ e a conformação_2 apresenta um V_TCN igual a $0,26$, a distância entre essas conformação antes de 1,0 Å, com a função $D_{1,2}TCN$ passa a ser de 1,33 Å. Ou seja, a distância entre essas duas conformações é aumentada devido ao baixo V_TCN da conformação_2. Entretanto, com o exemplo com a conformação_1 e conformação_4, a distância entre as conformações não se altera pois ambas as conformações apresentam um mesmo valor de V_TCN. Para cálculos de $D_{1,2}TCN$ entre conformações com altos valores de V_TCN a distância entre as mesmas diminui, o que aumenta a possibilidade de ambas conformações permanecerem em um mesmo grupo ao final da execução do algoritmo de agrupamento.

8.2.3.2 Função TCN_Mult2

Como uma variação da função TCN, definiu-se a função TCN_Mult2. Sendo a função RMS definida pelo valor de $D_{ab}RMS$ (Equação 8.1), a função TCN_Mult2 é definida por:

$$D_{a,b}TCN\_Mult2 = \frac{2 * D_{a,b}RMS}{V\_TCN_a + V\_TCN_b} \quad (8.5)$$

Onde, V_TCN_a é o total de contatos normalizado da conformação a e V_TCN_b é o valor total de contatos normalizados da conformação b.

A única diferença da função TCN_Mult2 para a TCN é o 2 no numerador multiplicando o valor de $D_{a,b}RMS$. Esse valor 2 foi escolhido devido aos valores de V_TCN do denominador da função no máximo poderem somar 2. Dessa forma, somente no caso extremo de ambas as conformações terem valores de V_TCN próximos ao máximo, o valor da $D_{a,b}TCN\_Mult2$ não modificaria o valor da $D_{ab}RMS$, em todos os outros casos, essa valor é influenciado pelos valores de V_TCN.

8.2.4 Funções Considerando a Matriz de Correlação entre os Resultados do LigPlot

Além das funções TCN e TCN_Mult2, desenvolveu-se outras 3 funções, chamadas funções $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$. Essas funções de similaridade fazem uso de uma matriz de correlação entre os resultados dos contatos estabelecidos entre as conformações. Essa matriz, chamada CORRELACAO, é determinada a partir da matriz de contatos como a exemplificada na Tabela 8.3. Na tabela CORRELACAO tem-se os valores de correlação de todas as conformações contra todas. Ou seja, trata-se de uma matriz 3.100 x 3.100, simétrica, mostrada em parte na Tabela 8.4. Os valores da matriz CORRELACAO são então utilizados para modificar o valor de distância $D_{a,b}RMS$ de forma a considerar a relação que há entre as conformações. Por exemplo, considerando que a função $D_{a,b}RMS$ entre duas conformações Conformação_1 e Conformação_2 seja igual a $1,0$Å, o valor de correlação entre ambas é 0,91 (Tabela 8.4). Na função $CORREL\_V1$

(Seção 8.2.4.1), divide-se o valor de $D_{a,b}RMS$ pelo valor de $CORRELACAO_{a,b}$, obtendo-se como resultado o valor de distância entre as conformações agora de $1,10\text{Å}$. Isso significa que os valores de CORRELACAO entre as conformações determinam o quanto a distância original RMS deverá ser aumentada para refletir a relação que há entre as conformações em relação aos totais de contatos estabelecidos. É importante ressaltar que nas funções $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ não estão sendo considerados somente os totais de contatos, mas também quais resíduos estão estabelecendo os contatos.

Tabela 8.4: Parte da matriz CORRELACAO gerada para a entrada THT.

	Conf. 1	Conf. 2	Conf. 3	Conf. 4	...	Conf. 3100
Conf. 1	1					
Conf. 2	0.91	1				
Conf. 3	0.91	0.99	1			
Conf. 4	0.31	0.37	0.38	1		
...	...	...	...	...	...	...
Conformação 3100	0.52	0.40	0.38	0.10	...	1

A partir da definição de que seriam utilizados esses valores da matriz CORRELACAO, foi necessário a definição de como esses valores seriam compostos com o valor de $D_{a,b}RMS$. Assim, foram desenvolvidas as 3 funções descritas a seguir.

8.2.4.1 Função CORREL_V1

A função mais simples é a $CORREL\_V1$:

$$D_{a,b}CORREL\_V1 = \begin{cases} \frac{D_{a,b}RMS}{0,3} & \text{se} \quad CORRELACAO_{a,b} \leq 0,3 \\ \frac{D_{a,b}RMS}{CORRELACAO_{a,b}} & \text{se} \quad CORRELACAO_{a,b} > 0,3 \end{cases} \tag{8.6}$$

O teste do valor de CORRELACAO é necessário porque alguns valores da matriz são negativos ou muito pequenos, próximos de zero. Esses valores, quando utilizados nos algoritmos de agrupamento, causavam erros na execução pois geravam valores de distância muito grandes, não tratáveis pelos algoritmos (com exceção dos algoritmos *Bayesian, SOM* e *K-means* que terminavam sua execução mesmo quando os valores da matriz CORRELACAO eram utilizados diretamente).

8.2.4.2 Funções $CORREL\_V2$ e $CORREL\_V3$

As funções $CORREL\_V2$ e $CORREL\_V3$ foram elaboradas para compor o valor de $D_{a,b}RMS$ de maneira diferente da $CORREL\_V1$. Foram testadas diversas funções para determinar essa composição e os melhores resultados foram obtidos com as fórmulas:

$$D_{a,b}CORREL\_V2 = \frac{10 * D_{a,b}RMS}{10^{CORRELACAO_{a,b}}} \tag{8.7}$$

$$D_{a,b}CORREL\_V3 = \frac{e * D_{a,b}RMS}{e^{CORRELACAO_{a,b}}} \tag{8.8}$$

Para melhorar o entendimento, o gráfico da Figura 8.8 mostra os valores de distância entre 2 conformações obtidos com as funções $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$, supondo sempre que o valor de $D_{a,b}RMS$ seja igual a $1,0$Å. Nesse gráfico os valores de distância $D_{a,b}CORREL\_V1$, $D_{a,b}CORREL\_V2$ e $D_{a,b}CORREL\_V3$ estão em Y e os valores da matriz CORRELACAO estão em X, onde foram exemplificados valores de correlação maiores do que 0,15. Como é possível ver com esse gráfico, a função $CORREL\_V2$ modifica mais drasticamente o valor de $D_{a,b}RMS$, enquanto que a função $CORREL\_V3$ modifica esse valor de forma mais sutil. O valor de $CORREL\_V1$ não muda até que a correlação atinja o valor de 0,3. Com essas 3 diferentes funções utilizando os valores da matriz CORRELACAO buscou-se analisar o impacto de uma função de similaridade linear ($CORREL\_V1$) e duas exponenciais ($CORREL\_V2$ e $CORREL\_V3$) nos algoritmos de agrupamento.

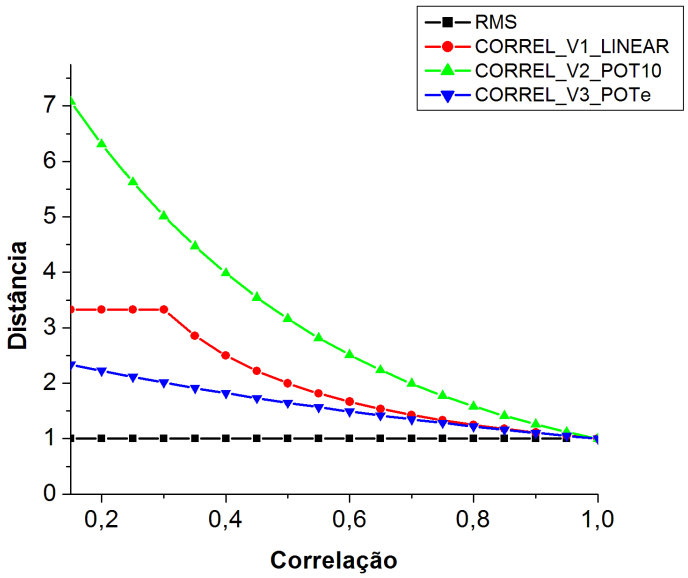


Figura 8.8: Gráfico de exemplo dos valores de distância entre duas conformações calculadas a partir das funções de similaridade RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$.

8.3 Resultados dos Experimentos de Agrupamento

Os experimentos com os 10 algoritmos de agrupamento foram divididos em 2 grupos para facilitar a discussão dos resultados obtidos:

- RMS X TCN: Apresenta os resultados para as funções RMS comparadas com os resultados das funções TCN e TCN_Mult2 que utilizam os totais de contatos normalizados estabelecidos entre as conformações do receptor e o THT ou THT+NADH;

- RMS X $CORREL$: Descreve os resultados de RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ para ambas as entradas THT e THT+NADH

8.3.1 Entrada para os Algoritmos de Agrupamento

Para os 2 grupos de experimentos foram utilizadas as seguintes conjuntos de átomos de entrada:

- ALL e 25_RES (descritas na Seção 8.1 deste capítulo): consideram os átomos de carbono-α dos 268 resíduos do receptor e dos Top 25 resíduos, respectivamente;

- 20_RES: considera os átomos de carbono-α dos 20 resíduos do receptor que aparecem estabelecendo ao menos 1 contato (HHB+NNB) com o THT. Esses 20 resíduos estão descritos na Tabela 8.2. Essa entrada somente foi considerada em experimentos com as funções de similaridade quando executadas com a entrada THT;

- 46_RES: são utilizados nesta entrada os átomos de carbono-α dos 46 resíduos do receptor que estabelecem contatos HHB ou NNB com o THT+NADH (Tabela 8.2). Essa entrada de algoritmo de agrupamento somente foi considerada em experimentos com funções de similaridade quando a entrada da mesma eram os resultados de THT+NADH;

8.3.2 Resultados TCN x RMS

Esse primeiro grupo de resultados correspondem aos experimentos com as funções RMS, TCN e TCN_Mult2 para os 10 algoritmos de agrupamento e subdividiu-se em 2 sub-grupos: os experimentos onde a entrada para as funções de similaridade TCN e TCN_Mult2 foram os resultados dos contatos do receptor-THT e os experimentos cuja entrada para as funções são os resultados de contatos do receptor-THT+NADH.

É importante diferenciar os conjuntos de átomos de entrada para os algoritmos de agrupamento, que correspondem aos átomos considerados para os cálculos de similaridade entre 2 conformações, das entradas para as funções de similaridade, que indicam qual tabela de contatos resultante do LigPlot foi aplicada em cada momento (THT ou THT+NADH).

As Figuras 8.9 e 8.10 mostram os resultados da métrica DBI para os 10 algoritmos, onde nas funções TCN e TCN_Mult2 foi aplicada a entrada THT+NADH. As Figuras 8.11 e 8.12 descrevem os resultados para da métrica pSF para essa mesma configuração de experimento. Essa configuração de experimento utilizou como conjunto de átomos de entrada ALL, 46_RES e 25_RES (a entrada 20_RES não foi considerada pois está relacionada aos resultados do LigPlot com o THT).

Nas Figuras 8.9, 8.10, 8.11 e 8.12 os gráficos de cada linha correspondem a um mesmo algoritmo, e as colunas são os diferentes conjuntos de átomos de entrada. Os resultados dessa mesma configuração de experimento porém com a entrada para as funções de similaridade sendo os resultados para o THT estão no Apêndice C. Nesta seção e nas próximas são descritos os resultados para os as funções de similaridade com entrada THT+NADH. Os resultados para as entradas THT são descritas sempre nos Apêndices desta Tese.

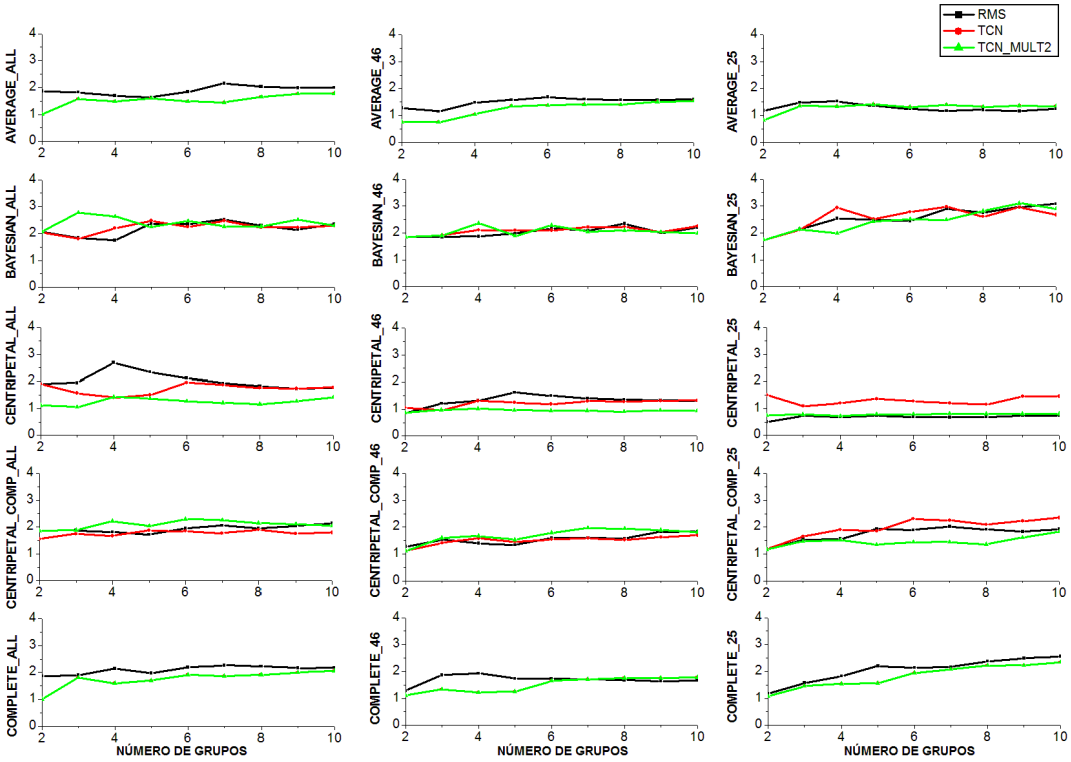


Figura 8.9: Resultado da métrica DBI para os algoritmos *Average, Bayesian, Centripetal, Centripetal_Comp* e *Complete* executados com as funções RMS, TCN e TCN_Mult2 com entrada THT+NADH.

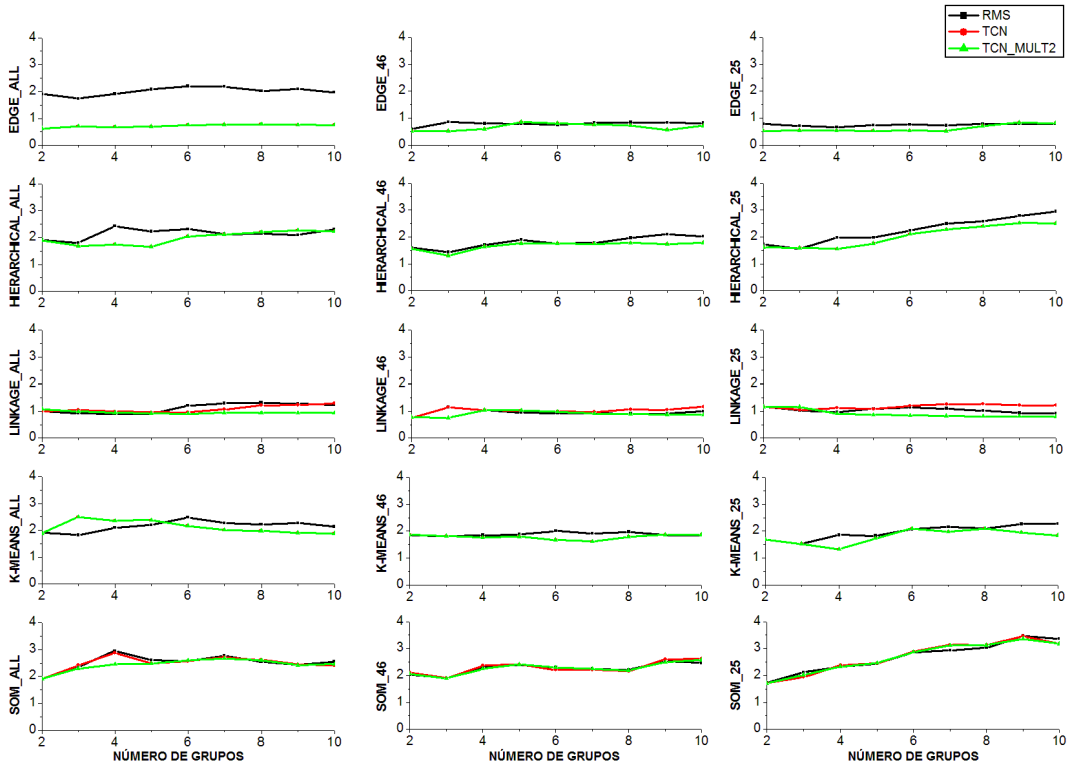


Figura 8.10: Resultado da métrica DBI para os algoritmos *Edge, Hierarchical, Linkage, K-means* e *SOM* executados com as funções RMS, TCN e TCN_Mult2 com entrada THT+NADH.

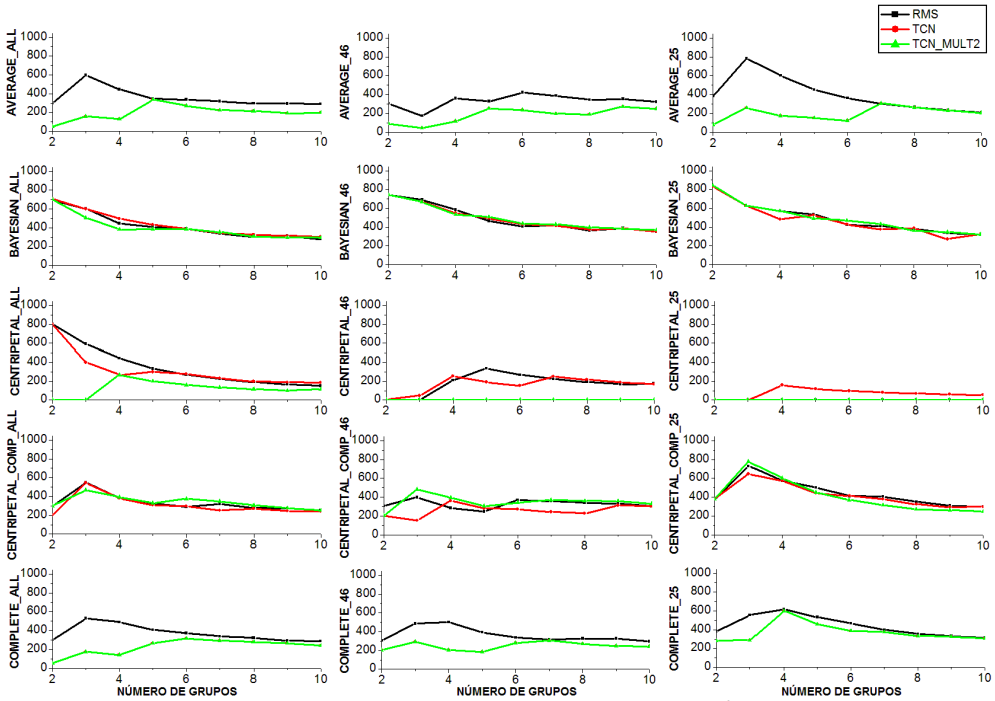


Figura 8.11: Resultado da métrica pSF para os algoritmos *Average*, *Bayesian*, *Centripetal*, *Centripetal_Comp* e *Complete* executados com as funções RMS, TCN e TCN_Mult2 com entrada THT+NADH.

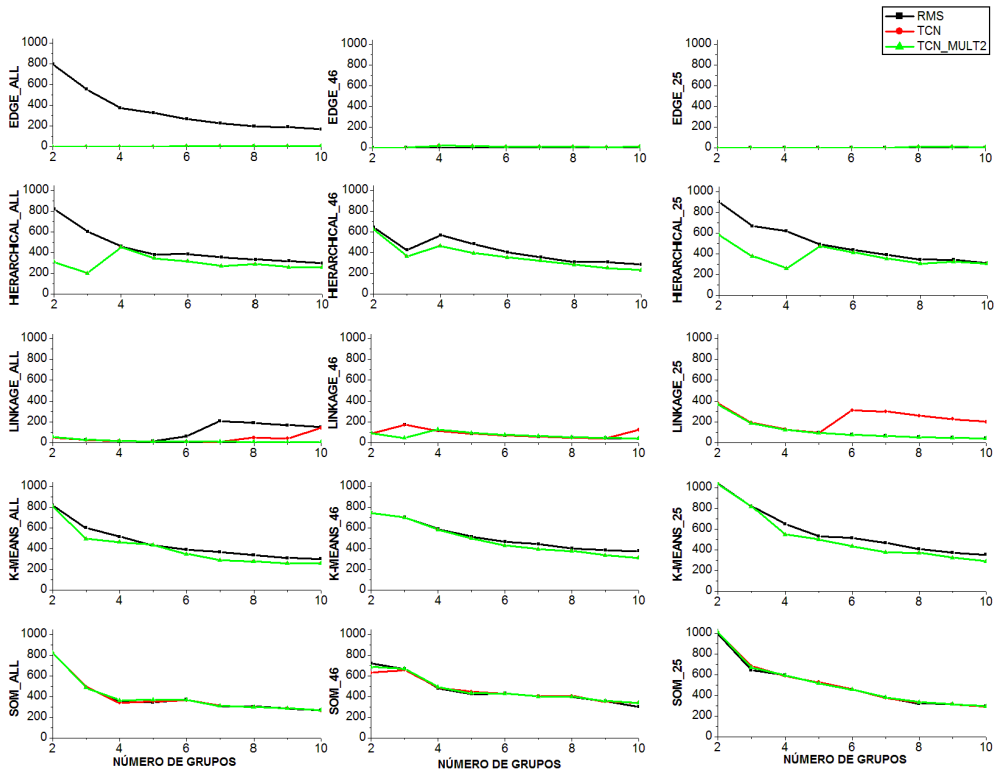


Figura 8.12: Resultado da métrica pSF para os algoritmos *Edge*, *Hierarchical*, *Linkage*, *K-means* e *SOM* executados com as funções RMS, TCN e TCN_Mult2 com entrada THT+NADH.

Sobre os valores de DBI, das Figuras 8.9 e 8.10, a primeira consideração é que para os algoritmos *Average, Complete, Edge, Hierarchical* e *K-means* os valores de TCN e TCN_Mult2 são iguais. Em relação a essa métrica, cujos menores valores correspondem a melhores resultados, para os algoritmos *Average, Complete* e *Edge* a função TCN_Mult2 apresenta valores melhores que a RMS para as diferentes entradas e os diferentes número de grupos. Os algoritmos *Hierarchical* e *K-means*, apesar de apresentarem valores aproximados, a função TCN_Mult2 é melhor em maior parte dos diferentes números de grupos e conjuntos de átomos testados. Os algoritmos *Linkage* e *SOM* apresentam valores muito próximos, sendo difícil a indicação de qual das funções têm melhores resultados. *Bayesian* tem muita variação entre qual das funções apresenta melhores valores de DBI. *Centripetal* tem melhores resultados para a função TCN_Mult2 para as entradas ALL e 46_RES, e para a função RMS com a entrada 25_RES. O contrário ocorre para o algoritmo *Centripetal_Comp*, em que TCN_Mult2 tem melhores resultados para 25_Res e para as entradas ALL e 46_Res os melhores valores de DBI variam entre as funções RMS e TCN.

Em relação a métrica pSF, onde maiores valores indicam melhores resultados, os algoritmos *Bayesian* e *SOM* apresentam valores muito aproximados, dificultando a verificação de qual das funções foi melhor. Os algoritmos *Average, Complete, K-means* e *Hierarchical* obtiveram melhores valores para a função RMS. *Edge* e *Linkage* foram os piores algoritmos para todas as funções analisadas, com exceção da execução do *Edge*, para a função RMS e entrada ALL, que apresenta bons resultados e do *Linkage* para 5 grupos ou mais, onde ALL gerou melhores resultados para RMS enquanto que 46_RES e 25_RES para as funções TCN e/ou TCN_Mult2. *Centripetal* obteve melhores valores de pSF para as entradas ALL e 46_RES, intercalando as funções RMS e TCN, enquanto que a entrada 25_RES somente gerou valores maiores do que zero para a função TCN. *Centripetal_Comp* apesar de gerar valores de pSF bem próximos para os átomos de entrada ALL e 25_RES, pode-se perceber alguns resultados melhores para TCN e TCN_Mult2. É importante ressaltar que os melhores valores de pSF, para a maioria dos algoritmos, aparecem para agrupamentos de até 5 grupos, após, os valores tendem a não variar muito.

Os resultados dos experimentos com as funções RMS X TCN para o THT (Apêndice C) foram muito parecidos com os resultados discutidos para THT+NADH. O padrão observado para a métrica DBI se manteve. Em relação a métrica pSF houveram alguns resultados diferentes, como por exemplo, para THT o algoritmo *Complete* e *Centripetal_Comp* apresentam resultados aproximados como o *SOM*, o *Centripetal* apresenta resultados ruins como o *Edge* e *Linkage*.

8.3.3 Resultados RMS X $CORREL$

O segundo grupo de resultados são executados com as funções RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$, para os 10 algoritmos de agrupamento com as funções de similaridade sendo executadas com as duas entradas: THT e THT+NADH. As Figuras 8.13 e 8.14 descrevem os resultados relacionados a métrica DBI, e as Figuras 8.15 e 8.16 aos valores de pSF, ambos considerando a entrada das funções como THT+NADH e como conjunto de átomos de entrada ALL, 46_RES e 25_RES.

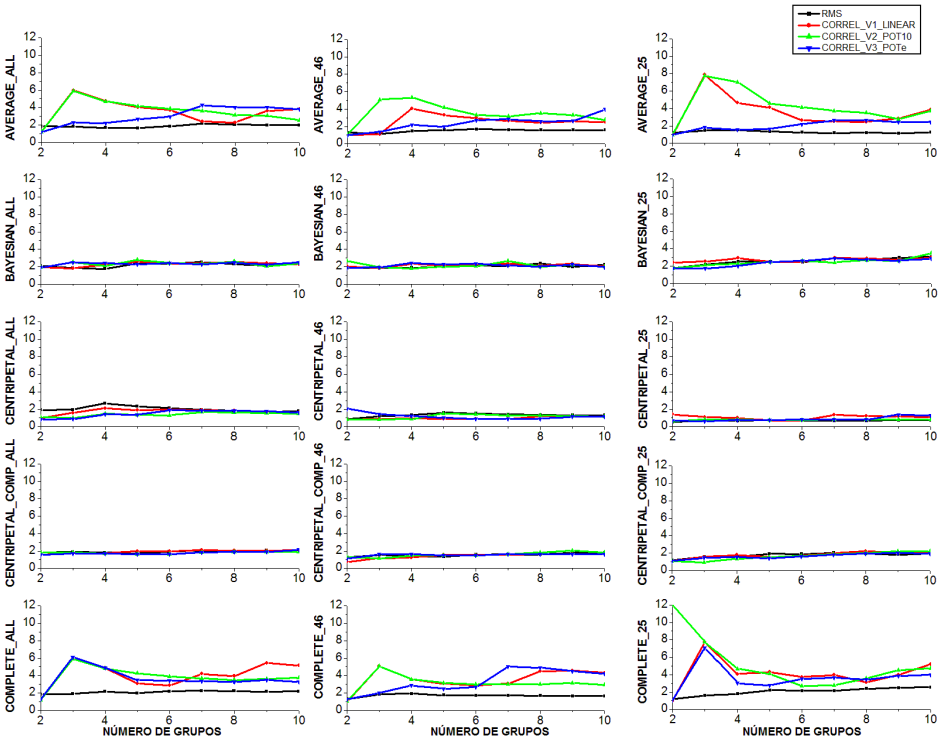


Figura 8.13: Resultado da métrica DBI para os algoritmos *Average*, *Bayesian*, *Centripetal*, *Centripetal_Comp* e *Complete* executados com as funções RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ com entrada THT+NADH.

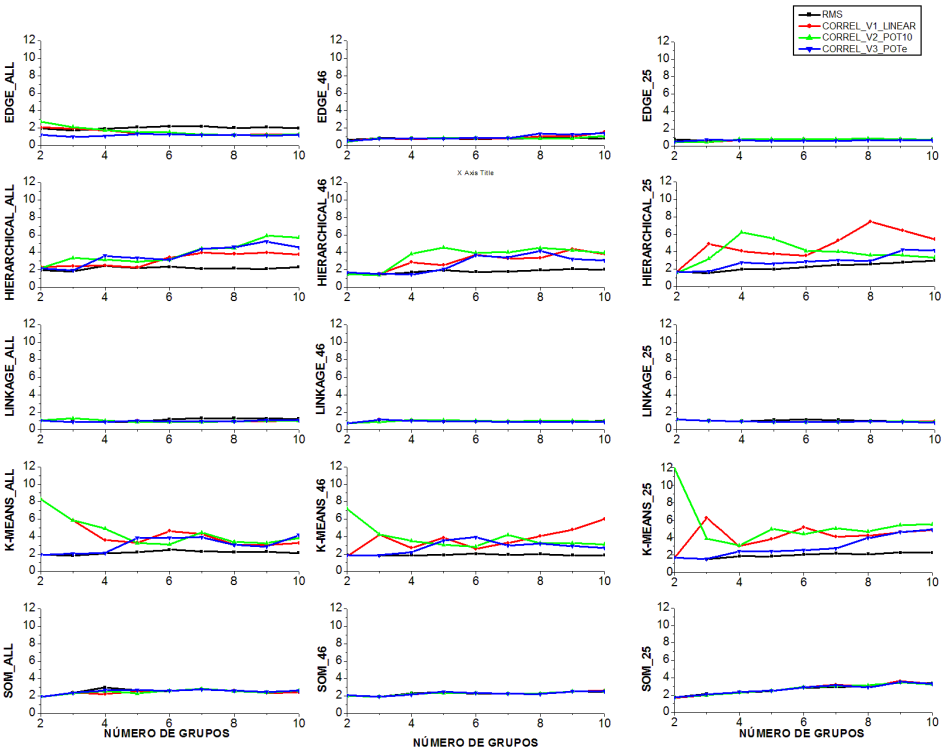


Figura 8.14: Resultado da métrica DBI para os algoritmos *Edge*, *Hierarchical*, *Linkage*, *K-means* e *SOM* executados com as funções RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ com entrada THT+NADH.

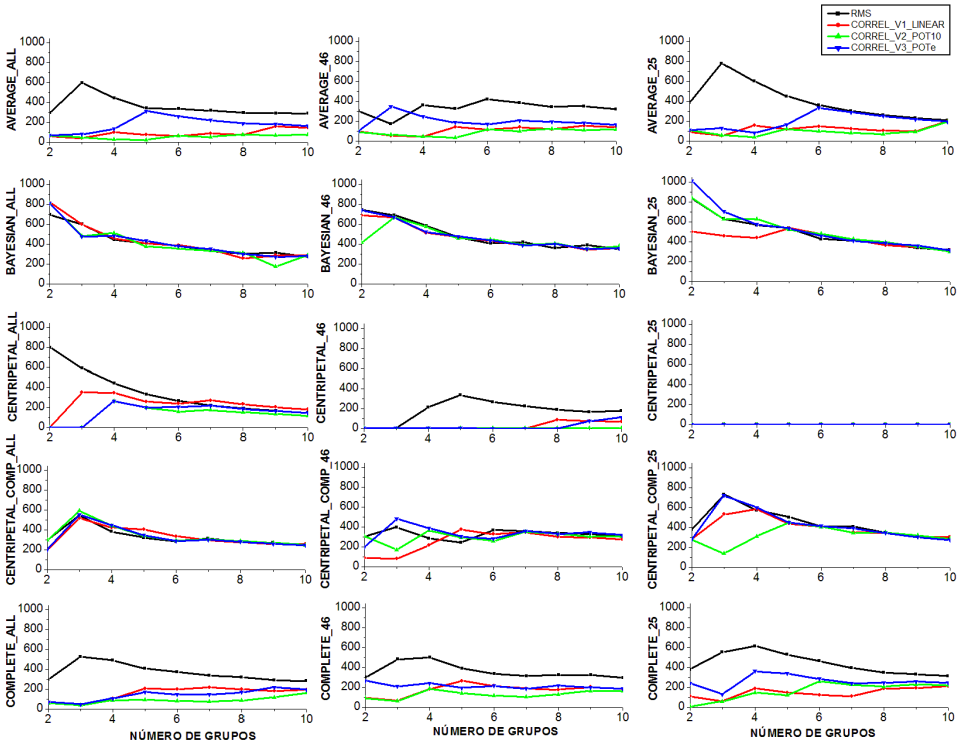


Figura 8.15: Resultado da métrica pSF para os algoritmos *Average*, *Bayesian*, *Centripetal*, *Centripetal_Comp* e *Complete* executados com as funções RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ com entrada THT+NADH.

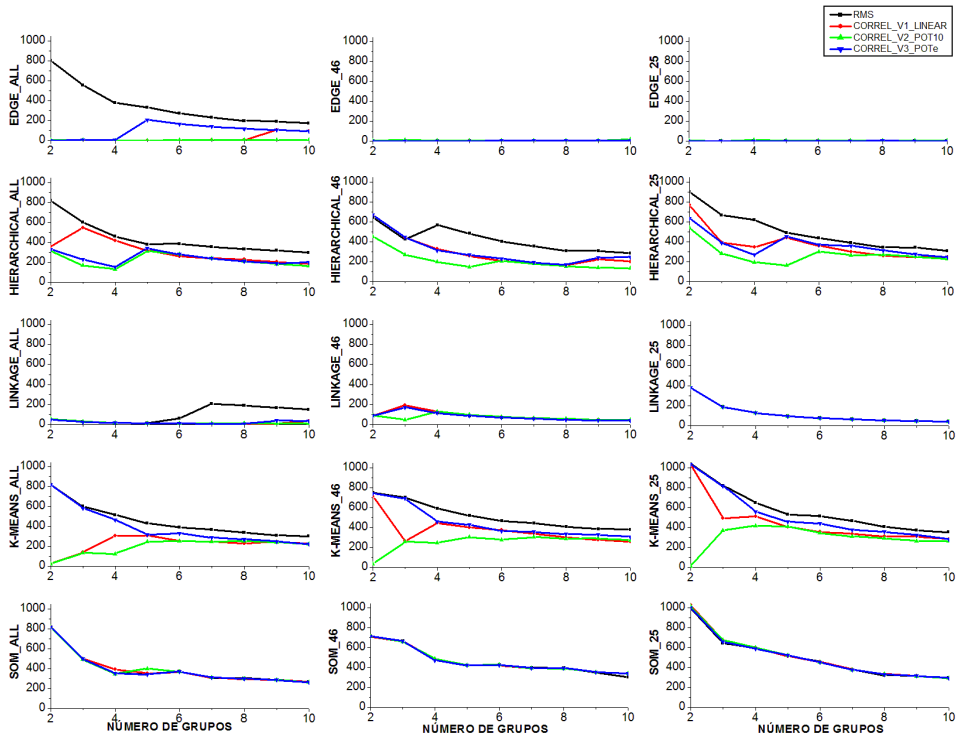


Figura 8.16: Resultado da métrica pSF para os algoritmos *Edge*, *Hierarchical*, *Linkage*, *K-means* e *SOM* executados com as funções RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ com entrada THT+NADH.

Os resultados apresentados nas Figuras 8.13 e 8.14 sobre a métrica DBI mostram que, para os algoritmos *Bayesian, Centripetal_Comp, Linkage* e *SOM* os valores são muito aproximados, não sendo possível a identificação das funções com melhores resultados. Para os algoritmos *Complete* e *Hierarchical* a função RMS é sempre melhor, independente dos átomos de entrada e do número de grupos. O mesmo ocorre para o *Average* e *K-means* , entretanto, para esses algoritmos, em alguns casos, a função $CORREL\_V3$ se aproxima dos valores de RMS. *Centripetal* e *Edge* são algoritmos cujos valores de DBI apesar de parecidos, mostram para a entrada ALL melhores valores para a função $CORREL\_V3$.

Os valores de pSF das Figuras 8.15 e 8.16 mostram que, novamente, os algoritmos *Bayesian* (ALL e 46_RES), *SOM* e Centripetal_Comp (ALL) têm valores muito aproximados, sendo possível destacar somente *Bayesian*-$CORREL\_V3$ com melhores valores para o conjunto de átomos de entrada 25_RES. Os algoritmos *Average, Complete, Hierarchical* e *K-means* têm melhores valores de pSF para a função RMS em todos os casos estudados, com exceção dos resultados para até 3 grupos com a função $CORREL\_V3$ que se aproximam de RMS. Os algoritmos *Edge* e *Linkage* apresentam os piores resultados entre os algoritmos, com exceção do *Edge*-RMS, que apresenta bons valores de pSF com entrada ALL e *Linkage*-$CORREL\_V3$ e entrada 25_RES. A função $CORREV\_V3$ apresenta melhores valores de pSF para os experimentos com Bayesian (25_RES) e Centripetal_Comp (46_RES e 25_RES).

Os resultados ruins em relação as métricas DBI e pSF para as funções $CORREL\_V1$ e $CORREL\_V2$ possivelmente foram causados por essas funções permitirem que sejam obtidos valores muito altos de distância entre 2 conformações (exemplo no Gráfico da Figura 8.8), o que pode prejudicar a execução dos algoritmos de agrupamento.

Os resultados para a mesma configuração de experimento mas para a entrada com THT estão no Apêndice D. De maneira geral, para as métrica DBI e pSF os resultados obtidos foram muito parecidos entre as entradas THT e THT+NADH.

8.4 Avaliações das Médias de Desvio Padrão (DP) de FEB Dentro de Cada Grupo

De acordo com Shao et al. [SHA07] as métricas de avaliação DBI e pSF são imperfeitas e o ideal é utilizá-las em conjunto e realizar uma inspeção visual nos resultados para conclusões sobre os melhores agrupamentos. Com base nos resultados apresentados, é difícil a indicação de um melhor algoritmo e de uma melhor função pois os melhores valores das métricas variam muito entre as diferentes configurações de experimentos de agrupamento executadas.

Por esses motivos, neste trabalho decidiu-se, além de avaliar os grupos com as métricas clássicas DBI e pSF implementadas em [SHA07], verificar os resultados dos agrupamentos na aplicação que os mesmos serão utilizados. O objetivo dos experimentos de agrupamento desde o início é utilizá-los em docagem molecular com o modelo FFR de receptor, de forma a acelerar experimentos desse tipo onde somente parte das conformações de cada grupo serão consideradas.

Para essa verificação dos resultados de agrupamentos na aplicação, em um primeiro momento foi

necessário a reexecução dos experimentos de docagem molecular (Experimentos Fase 2 - Seção 3.5.2 do Capítulo 3) pois, para utilização das conformações nos algoritmos de agrupamento foi necessária a sobreposição de todas as estruturas na primeira da DM, modificando as que foram utilizadas na Docagem-Fase 1.

Essa análise foi realizada para os resultados de docagem com os 4 ligantes: NADH, PIF, TCL e ETH. A partir destes, foram calculadas as médias de desvios padrão (DP) de FEB de cada agrupamento obtido para as diferentes configurações de experimentos. Os melhores valores nessa avaliação de média de DP são para as configurações com os menores valores. Estas menores médias de DP indicam que determinado agrupamento colocou conformações que apresentaram resultados de docagem mais similares em mesmos grupos. Por exemplo, tem-se experimentos de agrupamento quaisquer com 2 grupos cada, que obtiveram os seguintes valores de DP de FEB:

- experimento 1: Grupo 1 com DP de FEB de 0,5 Kcal/mol e Grupo 2 com DP de 1,5 kcal/mol. A média de DP para esse agrupamento é de 1,0 kcal/mol;
- experimento 2: Grupo 1 com DP de FEB de 2,5 Kcal/mol e Grupo 2 com DP de 1,5 kcal/mol. A média de DP para esse agrupamento é de 2,0 kcal/mol;

Neste exemplo simples, o melhor resultado é para o Experimento 1, uma vez que o mesmo apresenta menor variação de FEB entre seus grupos. Para realizar essa análise foi necessário cruzar as informações sobre FEB de cada conformação para cada ligante com os agrupamentos. Para isso, foi desenvolvido um pequeno Banco de Dados chamado *Docagem_Agrupamentos* que armazena todos os resultados de agrupamentos para as configurações executadas (6 tabelas, uma para cada função utilizada) e os resultados de docagem molecular de cada ligante (4 tabelas, uma para cada ligante). Assim, o cálculo de média de DP de FEB, utilizando os dados armazenados neste BD, foi aplicado para os 4 ligantes e:

- todas as funções de similaridade variando suas entradas entre THT e THT+NADH;
- número de grupos variando de 2 a 10;
- para os conjuntos de átomos de entrada 20_RES, 25_RES, 46_RES e ALL;
- para os 7 algoritmos de agrupamento com melhores resultados para as métricas DBI e pSF: *Average, Bayesian, Centripetal_Comp, Complete, Hierarchical, K-means* e *SOM*.

Os resultados dos agrupamentos com os algoritmos *Centripetal, Edge* e *Linkage*, especialmente para a métrica pSF, são ruins para quase todas as configurações de experimentos realizados, incluindo aqueles primeiros experimentos com de 10-100 grupos (Seção 8.1.1) e para de 2-20 grupos (Seção 8.1.2). Neste capítulo são descritos os resultados desta análise aplicado nos resultados de docagem para o PIF com ambas as entradas das funções de similaridade THT+NADH (Figuras 8.17 e 8.18) e THT (Figuras 8.19 e 8.20). Para os demais ligantes, os resultados estão apresentados no Apêndice E, considerando a entrada THT.

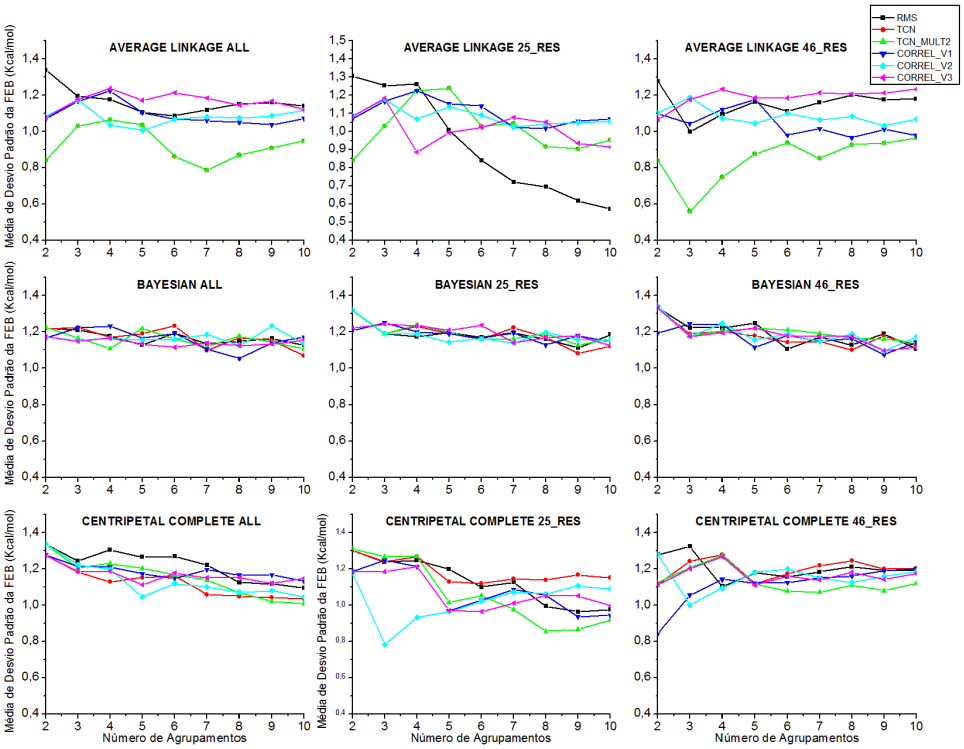


Figura 8.17: Média de desvio padrão de FEB para o ligante PIF com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT+NADH) para os algoritmos *Average, Bayesian* e *Centripetal_Comp* (ALL, 25_RES e 46_RES).

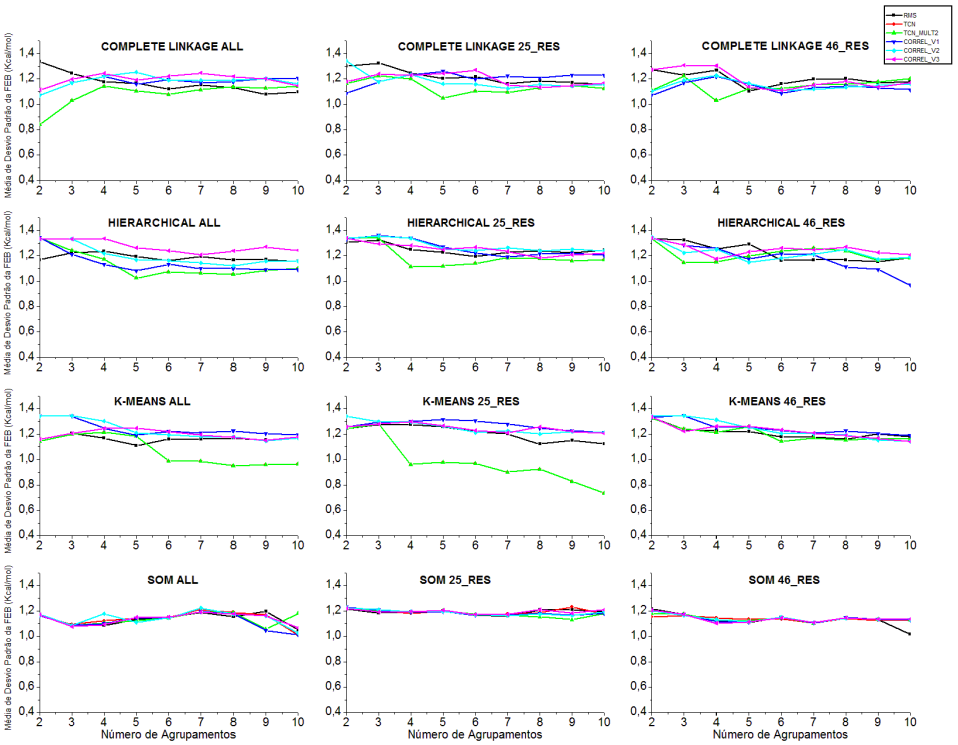


Figura 8.18: Média de desvio padrão de FEB para o ligante PIF com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT+NADH) para os algoritmos *Complete, Hierarchical, K-means* e *SOM* (ALL, 25_RES e 46_RES).

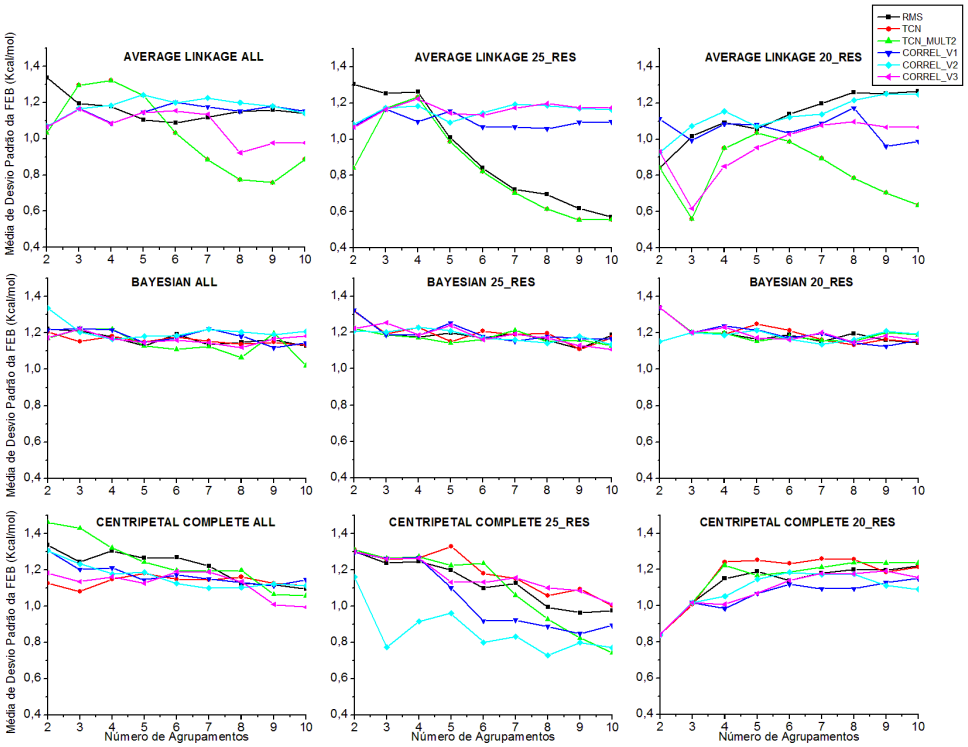


Figura 8.19: Média de desvio padrão de FEB para o ligante PIF com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT) para os algoritmos *Average, Bayesian* e *Centripetal_Comp* (ALL, 25_RES e 20_RES).

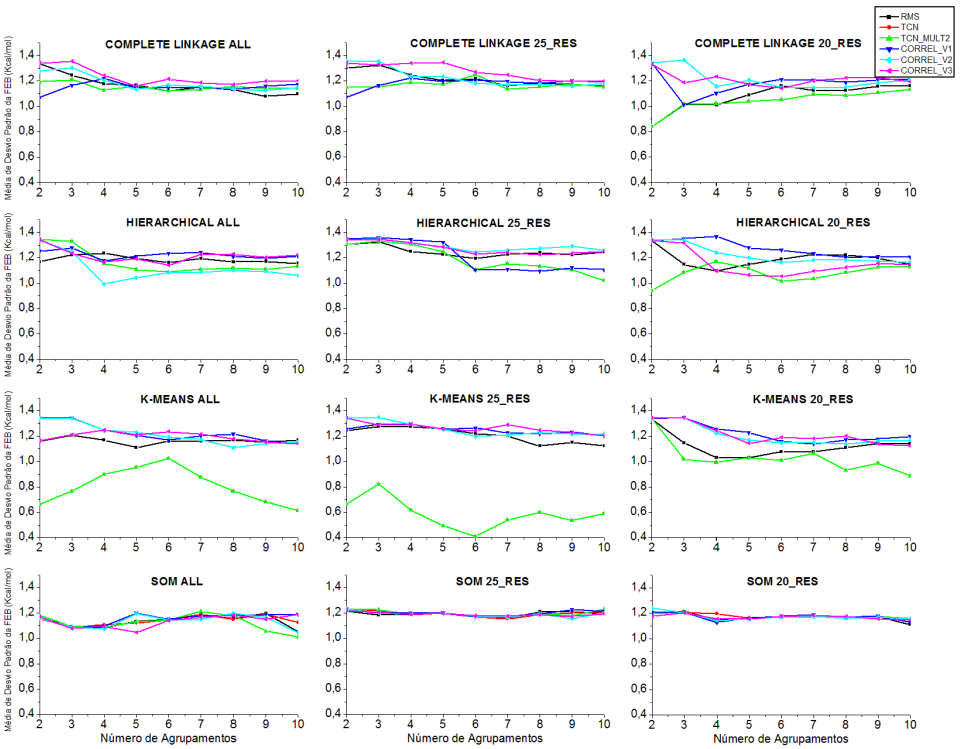


Figura 8.20: Média de desvio padrão de FEB para o ligante PIF com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT) para os algoritmos *Complete, Hierarchical, K-means* e *SOM* (ALL, 25_RES e 20_RES).

Os resultados descritos nas Figuras 8.17, 8.18, 8.19 e 8.20 são discutidos considerando todas as funções (RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$) para as diferentes configurações de experimentos:

- Novamente os algoritmos *Bayesian* e *SOM* apresentam valores aproximados para as diferentes funções. E isso ocorre para THT e THT+NADH. As funções com menores valores de DP variam muito para os diferentes número de grupos, não sendo possível indicar quais tem melhor resultado para esses algoritmos.

- O *Average*-THT+NADH apresenta menores valores de médias de DP para a função TCN_Mult2 para os átomos de entrada ALL e 46_RES, sendo superada em poucos casos pela função RMS. O mesmo algoritmo com a entrada para as funções THT também apresenta melhores valores para TCN_Mult2 na maioria dos casos, com exceção para ALL e 20_RES onde $CORREL\_V3$ ou RMS algumas vezes são melhores.

- Os melhores valores de DP para o algoritmo *Centripetal_Comp*-THT+NADH variam entre as funções $CORREL\_V2$ (para as 3 entradas), TCN (ALL) e TCN_Mult2 (25_RES e 46_RES). Para *Centripetal_Comp*-THT, $CORREL\_V2$ é melhor para as entradas ALL e 25_RES enquanto que $CORREL\_V1$ tem menores valores de DP de FEB para 46_RES.

- *Complete*-THT+NADH apresenta menores valores de DP para a função TCN_Mult2 (ALL e 25_RES) enquanto que para a entrada 46_RES há uma variação entre as funções. Menos valores de DP para *Complete*-THT ocorrem na maioria dos casos também para a função TCN_Mult2;

- O algoritmo *Hierarchical* apresenta os menores valores de DP de FEB para as funções $CORREL\_V1$, TCN_Mult2 e $CORREL\_V2$ para ambas as entradas THT+NADH e THT.

- De maneira geral, o algoritmo *K-means* apresenta menores valores de DP para a função TCN_Mult2 para a maioria das configurações de experimento realizadas com ambas entradas THT e THT+NADH.

Resumindo os resultados discutidos, juntamente com os resultados para os demais ligantes (Apêndice E) pode-se afirmar que a função TCN_Mult2 apresenta os menores valores de DP de FEB para as diferentes configurações de experimento. Além disso, também foi verificado que os melhores algoritmos foram *Average* e *K-Means*. Confirmou-se os resultados das métricas DBI e pSF em relação aos algoritmos Bayesian e SOM, que não tem seus resultados muito alterados independente da configuração de experimento executada. As funções relacionadas a correlação ($CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$) novamente não se mostram muito promissoras, como já havia sido verificado com as métricas DBI e pSF, com exceção de poucos casos.

8.5 Avaliação com o P-MIA

O P-MIA é um padrão de múltiplas instâncias autoadaptáveis, um padrão de dados para workflows científicos desenvolvido durante a Tese de Doutorado de Patrícia Hübler [HÜB10]. Esse trabalho foi realizado com o objetivo de contribuir com a redução da quantidade de docagens a serem executadas, via a definição de um padrão capaz de executar a seleção de conformações do receptor de forma dinâmica, onde não exista a necessidade de execuções exaustivas.

Para a utilização do P-MIA, a etapa preliminar consiste na execução do agrupamento, separando-se as conformações em diferentes grupos (não importando qual a configuração do experimento de agrupamento). Dessa forma, a cada conformação são relacionadas as seguintes informações: o grupo ao qual pertence, o lote e o *status*, que identifica a situação sobre o processamento da conformação, podendo ser Ativo (A), Finalizado (F), Descartado (D) ou Prioridade do grupo alterada (P) [HÜB10]. Essa informação de *status* é fundamental para determinar se uma conformação será considerada para docagem ou não (somente conformações com *status* (A) são processadas). O P-MIA também utiliza os valores de quantidade mínima de conformações (QM) a serem processados e o percentual da amostragem (PA) que formam cada lote, definidos pelo usuário.

Após a separação das conformações em grupos, o P-MIA subdivide os grupos em lotes. A quantidade de lotes é definida em tempo de execução baseada nos valores de QM e PA. Estudos descritos no trabalho de Patrícia [HÜB10] mostram que a análise de quantidades menores de dados (lotes) fornece melhores resultados. Um lote é formado pela quantidade de conformações indicadas por PA. As conformações de um determinado grupo que não entram em um lote formam o chamado lote residual (que pode ser processado ao final da execução com as conformações do lotes ou não).

A seguir, cada grupo é separado em em lotes e inicia-se a execução individual de cada conformação em um programa de um workflow científico. Como resultado obtém-se o chamado "Resultado Execução", que neste caso trata-se da FEB. Essa valor numérico é armazenado de alguma forma (arquivo, tabela em um Banco de Dados, etc.) e avaliado com base no intervalo [$Melhor\_valor$, $Pior\_valor$], que corresponde ao melhor e pior valor de FEB, respectivamente. As conformações cujo "Resultado Execução" se aproximam ou são menores do que o $Melhor\_valor$ são as conformações com maior probabilidade de sucesso [HÜB10].

Para o processamento dos lotes pelo workflow, são utilizados os seguintes parâmetros: $numero$, que corresponde a quantidade de conformações já processadas de um lote, $total\_resultado$, é o somatório dos resultados individuais de um grupo, $resultado\_snapshot$, é o valor final do processamento de determinada conformação, $total\_lote$, total de conformações de determinado lote, $melhor\_valor$ e $pior\_valor$, que correspondem ao melhor e pior valor a ser atingido. A partir desses parâmetros, o P-MIA calcula uma série de médias, como a média de FEB das conformações já processadas, o ponto médio de FEB do intervalo [$Melhor\_valor$, $Pior\_valor$] e a média amostral estimada, que considera as conformações ainda não processadas, utilizando para esse cálculo os valores de desvio padrão de FEB do grupo e do lote para as conformações já processadas. A fórmula da média amostral estimada está detalhada em [HÜB10] e é uma das principais contribuições

do modelo P-MIA e corresponde ao valor principal utilizado para indicar se determinado lote será descartado ou continuará sua execução.

Com o auxílio da Patrícia Hübler, que implementou algumas funcionalidades do P-MIA para os testes descritos em [HÜB10], o P-MIA foi aplicado a dois diferentes agrupamentos para a verificação se uma das funções de similaridade propostas nesta Tese apresenta ganho efetivo no processamento das conformações utilizados na docagem, comparando com a função RMS padrão. Como no trabalho [HÜB10] o objetivo era o padrão, para a verificação do mesmo, foram implementados os passos descritos acima com o auxílio de planilhas eletrônicas, sendo boa parte do trabalho feito manualmente. Por esse motivo, de todos os agrupamentos gerados nas mais diferentes configurações, a análise do P-MIA foi aplicada somente a um destes. O agrupamento escolhido foi com a função TCN_Mult2, executada com a entrada THT+NADH, para o algoritmo *K-means*, com conjunto de átomos de entrada ALL, com o total de grupos igual a 6. A mesma configuração de agrupamento foi aplicada à seleção dos resultados com a função RMS. O ligante PIF foi escolhido por ter sido um dos ligantes testados em [HÜB10].

Antes da aplicação do P-MIA os dados foram preparados, onde para ambos agrupamentos foram associados os valores de FEB a suas respectivas conformações, dentro dos diferentes lotes e dos diferentes grupos, utilizando para isso o BD *Docagem_Agrupamentos*. Após a separação das conformações em lotes e a associação dos resultados de FEB obtidos, o P-MIA calcula os valores de média e média estimada para a determinação de continuidade ou não do processamento.

A Tabela 8.5 contém o total de conformações que compõem cada um dos grupos, gerados para as configurações de agrupamento K-means-ALL-6_grupos-THT+NADH com as funções de similaridade RMS e TCN_Mult2. Esses grupos foram então divididos em lotes com QM=50 e PA=30%. Para a identificação de qual seria o percentual a ser utilizado para a definição de continuidade ou descarte de um lote, foram analisados valores de 20%, 30%, 50%, 70% e 80%.

Tabela 8.5: Quantidade de conformações em cada grupo, gerados pelo algoritmo *K-means* com as funções de similaridade RMS e TCN_Mult2

Grupos	Quantidade de conformações RMS	Quantidade de conformações TCN_Mult2
0	291	293
1	474	379
2	801	1
3	507	1.011
4	522	807
5	505	609

As Figuras 8.21, 8.22 mostram exemplos das análises realizadas para 30% onde as colunas referem-se a (1) lote de cada grupo (C_L); (2) quantidade total de conformações do lote (Quant); (3) média aritmética de FEB das conformações do lote até o momento da análise (M20%, M30%, M40%, M50%, M70%, M80%); (4) média estimada de FEB das conformações restantes até o momento da análise (E20%, E30%, E40%, E50%, E70%, E80%); (5) quantidade de conformações processadas até o momento (Proc); (6) quantidade total de conformações processadas (ProcFinal)

e (7) quantidade de conformações não processadas (Ganho).

C_L	Quant	M20%	E20%	Proc	M30%	E30%	Proc	M50%	E50%	Proc	M70%	E70%	Proc	M80%	E80%	Proc	ProcFinal	Ganho
C0_L0	87	-10,41	-10,49	17	-10,31	-10,45	26	-10,31	-10,58	44	-10,55	-10,7	61	-10,5	-10,64	70	87	0
C0_L1	61	-11,94	-12,18	12	-11,35	-11,8	18	-11,07	-11,48	31	-11,02	-11,24	43	-10,77	-10,94	49	61	0
C0_L2	50	-8,92	-9,32	10	-8,87	-9,16	15										15	35
C0_L3	93	-9,97	-10,69	19	-9,86	-10,42	28	-9,5	-9,88	47							47	46
C1_L0	142	-8,87	-9,2	28	-8,73	-9,11	43										43	99
C1_L1	100	-8,84	-9,01	20	-9,16	-9,44	30										30	70
C1_L2	70	-9,22	-9,57	14	-9,22	-9,49	21										21	49
C1_L3	50	-8,24	-8,83	10	-8,29	-8,72	15										15	35
C1_L4	50	-7,97	-8,35	10	-8,07	-8,36	15										15	35
C1_L5	63	-8,85	-9,2	12	-9,36	-9,79	19										19	44
C2_L0	240	-9,96	-10,54	48	-9,51	-10,02	72										72	168
C2_L1	168	-11,08	-11,4	34	-11,07	-11,31	50	-11,18	-11,36	84	-11,02	-11,16	118	-10,96	-11,01	134	168	0
C2_L2	118	-11,42	-11,63	24	-11,14	-11,33	35	-11,04	-11,22	59	-11,08	-11,22	83	-11,02	-11,05	94	118	0
C2_L3	83	-10,62	-10,87	17	-10,89	-11,11	25	-10,81	-11,01	42	-10,76	-10,82	58	-10,69	-10,7	66	83	0
C2_L4	57	-9	-9,3	11	-9,29	-9,57	17										17	40
C2_L5	50	-9,35	-9,5	10	-9,61	-9,81	15										15	35
C2_L6	85	-9,56	-9,76	17	-9,98	-10,31	26										26	59
C3_L0	152	-10,18	-10,73	30	-10,36	-10,86	46	-10,54	-10,86	76	-10,7	-10,85	106	-10,7	-10,84	122	152	0
C3_L1	106	-9,29	-9,56	21	-9,65	-10,02	32										32	74
C3_L2	75	-10,67	-11,36	15	-10,45	-11,03	23	-10,4	-10,83	38	-10,21	-10,49	53	-10,15	-10,29	60	60	15
C3_L3	52	-10,64	-11,06	10	-10,33	-10,92	16	-10,76	-11,55	26	-10,68	-10,8	36	-10,79	-11,01	42	52	0
C3_L4	50	-10,61	-11,22	10	-10,55	-11,12	15	-10,35	-11,39	25	-10,55	-10,75	35	-10,56	-10,69	40	50	0
C3_L5	72	-9,8	-10,33	14	-9,48	-9,99	22										22	50
C4_L0	157	-9,48	-9,88	31	-9,93	-10,31	47										47	110
C4_L1	110	-7,58	-7,96	22	-7,82	-8,13	33										33	77
C4_L2	77	-9,32	-9,63	15	-9,39	-9,73	23										23	54
C4_L3	54	-8,24	-8,5	11	-8,49	-8,72	16										16	38
C4_L4	50	-9,03	-9,47	10	-9,07	-9,42	15										15	35
C4_L5	75	-8,65	-8,86	15	-8,61	-8,78	22										22	53
C5_L0	152	-10,18	-10,55	30	-10,39	-10,76	46	-10,34	-10,57	76	-10,33	-10,44	106	-10,29	-10,4	122	152	0
C5_L1	106	-10,03	-10,35	21	-10,28	-10,59	32	-10,11	-10,3	53							53	53
C5_L2	74	-9,24	-9,55	15	-9,4	-9,62	22										22	52
C5_L3	52	-9,39	-9,7	10	-9,14	-9,53	16										16	36
C5_L4	50	-9,28	-9,66	10	-9,22	-9,54	15										15	35
C5_L5	71	-8,77	-8,88	14	-8,7	-8,79	21										21	50

Figura 8.21: Análise dos resultados com 30% das conformações processadas. Resultados da função de similaridade RMS.

C_L	Quant	M20%	E20%	Proc	M30%	E30%	Proc	M50%	E50%	Proc	M70%	E70%	Proc	M80%	E80%	Proc	ProcFinal	Ganho
C0_L0	88	-10,42	-10,6	18	-10,31	-10,42	26	-10,31	-10,52	44	-10,55	-10,73	62	-10,5	-10,54	70	88	0
C0_L1	62	-12,03	-12,23	12	-11,33	-11,9	19	-11,04	-11,35	31	-11,02	-11,12	43	-10,64	-10,85	50	62	0
C0_L2	50	-8,6	-8,79	10	-8,63	-8,76	15										15	35
C0_L3	93	-10,13	-10,79	19	-9,96	-10,49	28	-9,59	-9,97	47							47	46
C1_L0	114	-10,53	-10,99	23	-10,21	-10,7	34	-9,92	-10,27	57							57	57
C1_L1	80	-10,44	-10,98	16	-10,16	-10,61	24	-9,56	-9,89	40							40	40
C1_L2	56	-9,73	-10,06	11	-9,42	-9,76	17										17	39
C1_L3	50	-9,74	-10,27	10	-9,6	-10,02	15										15	35
C1_L4	79	-10,89	-11,15	16	-10,93	-11,17	24	-10,99	-11,23	40	-11,05	-11,1	55	-11,05	-11,09	63	79	0
C2_L0	1	-10,25															1	0
C3_L0	303	-10,49	-11,01	61	-10,82	-11,23	91	-10,62	-10,93	152	-10,46	-10,63	212	-10,33	-10,44	242	303	0
C3_L1	212	-10,66	-11,22	42	-10,76	-11,26	64	-10,72	-11,06	106	-10,85	-11,02	148	-10,86	-11	170	212	0
C3_L2	149	-10,89	-11,17	30	-11,05	-11,3	45	-10,96	-11,17	75	-10,98	-11,06	104	-10,91	-10,96	119	149	0
C3_L3	104	-10,96	-11,23	21	-10,87	-11,05	31	-10,71	-10,87	52	-10,69	-10,81	73	-10,49	-10,56	83	104	0
C3_L4	73	-9,28	-9,61	15	-9,57	-9,85	22										22	51
C3_L5	50	-9,32	-9,51	10	-9,49	-9,66	15										15	35
C3_L6	50	-10,31	-10,62	10	-10,29	-10,56	15	-10,49	-10,66	25	-10,58	-10,67	35	-10,6	-10,68	40	50	0
C3_L7	70	-10,2	-10,54	14	-10	-10,29	21										21	49
C4_L0	242	-8,34	-8,7	48	-8,38	-8,7	73										73	169
C4_L1	169	-8,57	-8,89	34	-8,65	-8,93	51										51	118
C4_L2	119	-9,34	-9,59	24	-9,17	-9,39	36										36	83
C4_L3	83	-9,36	-9,54	17	-9,35	-9,54	25										25	58
C4_L4	58	-7,92	-8,55	12	-7,82	-8,2	17										17	41
C4_L5	50	-7,94	-8,27	10	-7,7	-7,98	15										15	35
C4_L6	86	-8,96	-9,33	17	-8,94	-9,31	26										26	60
C5_L0	183	-10,36	-10,7	37	-10,1	-10,38	55										55	128
C5_L1	128	-9,23	-9,57	26	-9,32	-9,55	38										38	90
C5_L2	90	-8,87	-9,13	18	-8,82	-9,02	27										27	63
C5_L3	63	-10,16	-10,61	13	-9,76	-10,19	19										19	44
C5_L4	50	-9,18	-9,43	10	-9,23	-9,41	15										15	35
C5_L5	95	-9,1	-9,45	19	-9,14	-9,46	29										29	66

Figura 8.22: Análise dos resultados com 30% das conformações processadas. Resultados da função de similaridade TCN_Mult2.

A análise dos resultados de médias aritmética e estimada de FEB se inicia quando 20% das conformações já foram processadas. Após, é determinado qual é o *status* de cada lote. Se, ao analisar os valores de média aritmética e média estimada de determinado lote, ambos os valores forem piores do que o valor médio utilizado como parâmetro, o lote é Descartado (D). Os lotes

com essa característica são sombreados nas Figuras 8.21, 8.22. São geradas tabelas como as exemplificadas nas Figuras 8.21, 8.22 para a avaliação do ganho com 20%, 30%, 50%, 70% e 80% das conformações processadas utilizando a abordagem do P-MIA para a determinação do *status* de cada lote a medida que vai avançando o processamento dos mesmos.

Considerando as Figuras 8.21, 8.22 onde as análises começaram a ser feitas quando 30% das conformações de cada lote haviam sido processadas, para a função RMS, 1.446 conformações foram descartadas, ou seja, um ganho de 47%. A análise com os resultados da função TCN_Mult2 1.376 conformações foram descartadas, o que corresponde a um ganho de 44%. O gráfico da Figura 8.23 mostra o ganho obtido a medida que as análises com 20%, 30%, 50%, 70% e 80% foram sendo realizadas.

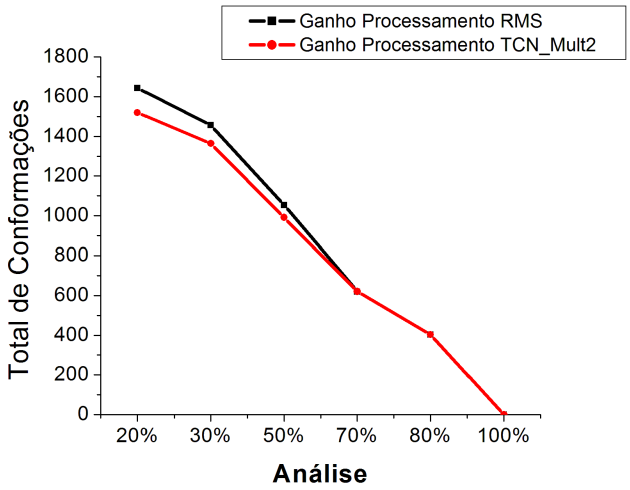


Figura 8.23: Ganho (total de conformações descartadas) obtido à medida em que as análises foram realizadas.

No gráfico da Figura 8.23 pode-se observar que a melhor alternativa é de iniciar a análise o quanto antes, ou seja, com 20% das conformações processadas, onde para a função RMS 1.648 conformações foram descartadas, o que corresponde a um ganho de 53% e para a função TCN_Mult2, 1.521 conformações não foram processadas, o que equivale a um ganho de 49%. Para aprofundar o estudo do ganho obtido com o uso do P-MIA e das funções de similaridade, foi verificado se as conformações com melhores resultados foram contempladas, ou seja, se foram processados a medida que as análises eram realizadas. O gráfico da Figura 8.24 apresenta essa análise, onde Melhores 10% referem-se as 310 conformações em que no experimento exaustivo descrito na Seção 3.5.2 do Capítulo 3 apresentaram os melhores resultados de FEB.

Como pode-se ver na Figura 8.24, somente com 20% das conformações processadas, para função RMS, 239 das 310 conformações foram contempladas (77%) e com a função TCN_Mult2, 254 (82%). Os resultados do processamento com o P-MIA mostram que este padrão de workflow utilizando em conjunto com os resultados dos experimentos de agrupamento apresenta um ganho muito importante na execução de simulações de docagem molecular com o modelo FFR.

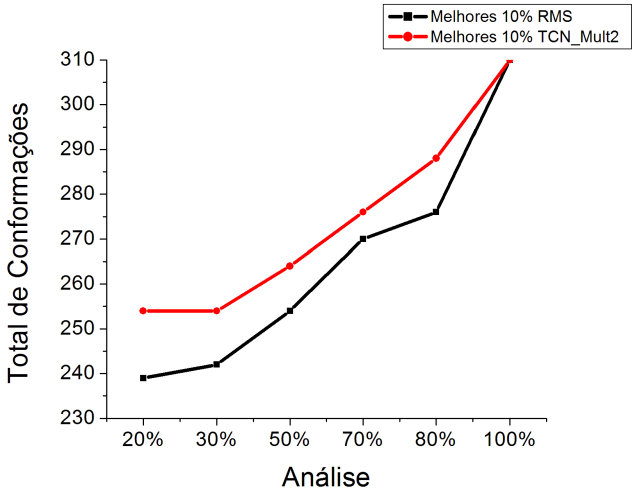


Figura 8.24: Avaliação do número de conformações das Melhores 10% contempladas a cada análise.

8.6 Considerações Finais

Este capítulo apresentou todos os experimentos de agrupamento executados com diferentes configurações. As funções de similaridade desenvolvidas são descritas e seus resultados são comparados com a função original. Ao final do capítulo foi realizada uma análise dos resultados obtidos utilizando para isso o padrão de *workflow* P-MIA para efetivamente acelerar as simulações de docagem molecular com o receptor flexível.

Os primeiro experimentos executados somente com a função RMS mostraram que mais do que 20 grupos não causavam modificações nas métricas de avaliação dos grupos (DBI e pSF). Para um estudo mais detalhado, foram executados os experimentos de 2-20 grupos, mas variando de 1 em 1. Neste estudo, decidiu-se que mais do que 10 grupos não eram necessários e, além disso, verificou-se que os dois conjuntos de átomos testados (ALL e 25_RES) apresentavam resultados aproximados, não sendo possível indicar qual era o melhor.

Para o desenvolvimento das novas funções de similaridade foram utilizados os resultados do processamento com o programa LigPlot, que analisa os contatos estabelecidos entre determinado complexo receptor-ligante. As análises com o LigPlot foram feitas com duas entradas diferentes, considerando as conformações do receptor e o substrato THT e o receptor com o THT+NADH. A partir destes, foram desenvolvidos 5 funções de similaridade divididas em 2 grupos: as funções TCN e TCN_Mult2, que utilizam os valores de totais de contatos entre receptor-ligante e as funções $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ que consideram matrizes de correlação entre as conformações obtidas a partir de como cada conformação estabeleceu seus contatos. Dessa forma, foram executados experimentos com as seguintes configurações:

- funções de similaridade: RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$;
- entrada paras as funções de similaridade: THT e THT+NADH;

- algoritmos: *Average, Bayesian, Centripetal, Centripetal_Comp, Complete, Edge, Hierarchical, Linkage, K-means* e *SOM*;
- número de grupos: de 2, 3, 4, 5, 6, 7, 8, 9 e 10;
- conjuntos de átomos de entrada: ALL, 25_RES, 20_RES e 46_RES.

A partir dos resultados obtidos para todos esses experimentos, incluindo a análise de média de DP de FEB dentro de cada grupo, pode-se concluir:

- É muito difícil o desenvolvimento de funções que geram bons resultados para todos os testes executados, já que muitas variações foram realizadas, trata-se de muitos e diferentes algoritmos, como diferentes conjuntos de átomos de entrada para as diferentes funções de similaridade.
- Não houveram muitas diferenças entre as entradas THT e THT+NADH. Acredita-se que isso se deve ao fato de mesmo o LigPlot tendo sido executado com diferentes entradas, o padrão de contatos estabelecidos entre as conformações e THT ou THT+NADH se manteve o mesmo, principalmente ao comparar-se os valores de totais normalizados.
- Considerando a comparação RMS X TCN, de maneira geral, considerando somente DBI, as funções TCN e/ou TCN_Mult2 apresentam ou valores aproximados ou melhores valores do que RMS para a maioria dos algoritmos, diferentes números de grupos e diferentes conjuntos de átomos de entrada.
- As funções $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ não se mostram muito promissoras nem em relação as métricas nem em relação as médias de DP de FEB dos grupos.
- Em relação aos algoritmos de agrupamento:
  - os melhores valores de métricas DBI e pSF assim como de médias de DP de FEB dos grupos foram obtidos em sua maioria com os algoritmos *Average* e *K-means*. Estes mesmos algoritmos foram indicados no trabalho de [SHA07] como de melhores resultados;
  - os algoritmos Edge, Linkage e Centripetal apresentaram valores de pSF ruins para quase todos as configurações, inclusive para a função RMS. O que acontece nesses algoritmos é que muitas vezes a maioria das conformações ficam em um único grupo, e os outros grupos ficam com somente 1 elemento. Esse problema já havia sido relatado para o Edge e Linkage no trabalho de [SHA07];
  - SOM e Bayesian foram algoritmos cujas diferentes configurações dos experimentos não afetaram muito seus resultados, não sendo possível a verificação nestes, de qual função de similaridade era melhor. Inclusive na análise com as médias de DP para esses algoritmos elas não variam muito. Em [SHA07] é descrito que o ideal para esses algoritmos é

a execução dos mesmos no mínimo 5 vezes para cada configuração, obtendo a média das métricas de cada execução. Acredita-se que, por isso não ter sido realizado, esses resultados não apresentaram a diferença esperada.

- Segundo Shao et al. [SHA07] as métricas de avaliação DBI e pSF são imperfeitas. Assim, não é possível, somente com base nos valores destas indicar quais são as configurações de agrupamento mais promissoras. Por esse motivo foram feitas as análises de médias de DP de FEB dos grupos, que mostraram que as funções desenvolvidas nesta Tese tendem a diminuir os valores de DP, o que indica que houve melhora nos agrupamento em relação a aplicação dos mesmos em docagem molecular. Em especial para a função de similaridade TCN_Mult2, que se mostrou entre as funções desenvolvidas ser a mais promissora. A função RMS aparece em poucos resultados desta análise de média de DP com melhores valores, e isso ocorre para os 4 ligantes estudados;

Em relação ao tempo de processamento utilizando a função RMS e as funções desenvolvidas TCN e TCN_Mult2, não houve aumento nesse tempo, tendo os algoritmos despendidos o mesmo tempo. As funções $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ em relação a função RMS tiveram seu tempo de execução aumentado somente no início de cada experimento, para a leitura da matriz CORRELACAO. A execução do LigPlot para a geração dos dados utilizados nas funções desenvolvidas despendeu em torno de 8 horas para as duas entradas, em um computador Core2Duo, 2GB RAM, mas esse procedimento só é necessário de ser executado uma vez.

A análise com o P-MIA mostrou bons resultados de ganhos para ambas as funções RMS e TCN_Mult2. Apesar da função RMS apresentar em torno de 5% mais ganho do que a TCN_Mult2, em relação ao número de conformações descartadas a cada porcentagem de análise (a vantagem se mantém até aproximadamente as análises com 60%), a função TCN_Mult2 contempla maior número de conformações das 10% melhores, e a diferença sobre a função RMS se mantém para todas as análises feitas em torno de 5%. Ou seja, mesmo que TCN_Mult2 tenha descartado um número menor de conformações, contemplou conformações comprovadamente mais promissoras.

A análise com o padrão P-MIA mostra um ganho de processamento muito importante, utilizando tanto a função RMS quanto a função TCN_Mult2, uma vez que com somente 20% das conformações processadas, houve ganhos de aproximadamente 50%, o que possibilitaria a execução dos experimentos de docagem em um tempo consideravelmente mais reduzido. Além do mais, com os mesmos 20% de processamento, 77% (RMS) e 82% (TCN_Mult2) das melhores conformações foram consideradas. Ou seja, com 20% do tempo de um experimento exaustivo, 80% das melhores conformações já foram consideradas. Isso significa por exemplo que, um experimento que antes precisava em torno de 12 horas em um computador QuadCore com 8GB RAM, utilizando o P-MIA com os agrupamentos gerados ele despende aproximadamente $1/5$ desse tempo, 2 horas e 24 minutos.

Ainda são necessárias análises com o P-MIA para outras das configurações de agrupamento executadas, mas para isso há a necessidade da implementação do padrão em um workflow cientifico,

o que já está sendo realizado por um aluno de Mestrado da PUCRS. Com os resultados das análises do P-MIA apresentados nesta Tese e em [HÜB10] pode-se concluir que sua utilização é muito promissora para a redução do tempo de execução das simulações de docagem molecular com o modelo FFR, mantendo as características dessa flexibilidade.

9. TRABALHOS RELACIONADOS

Este capítulo apresenta os trabalhos relacionas ao desenvolvido nesta Tese, que incluem:

- um trabalho sobre um Banco de Dados para triagem virtual de compostos, que pode ser comparado ao FReDD;
- trabalhos relacionados a execução de docagem molecular com o receptor flexível e seleção de conformações; e
- trabalhos relacionados ao agrupamento de trajetórias de DM.

9.1 Banco de Dados para RDD ou Docagem Molecular

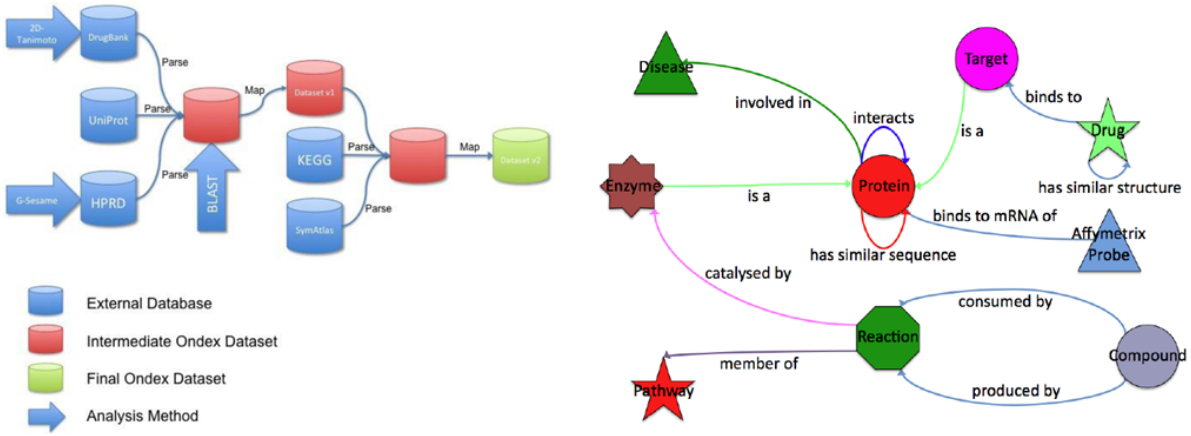
9.1.1 Um Banco de Dados para Triagem Virtual de Compostos [COC10]

Nesse trabalho é apresentado um BD desenvolvido utilizando a plataforma de integração Ondex [KOH06] para relacionar dados sobre a descoberta de novos candidatos a fármacos *in-silico*. A motivação dos autores em desenvolver essa base de dados é de encontrar exemplos de moléculas que devem ter uma proposta terapêutica adicional as já existentes para determinado alvo.

A plataforma Ondex utilizada para essa integração de dados endereça esse problema representando os muitos tipos de dados com nodos, que contém o que os autores chamam de *Concepts* e conexões representam o que os autores definem como *Relations*. A integração dos dados é então representada como uma rede de nodos (*Concepts*) interconectados (*Relations*) na forma de um grafo que pode ser enriquecido semanticamente com metadados, dessa forma, múltiplas fontes de informação são colocadas significativamente juntos em um grafo [COC10].

Nesse trabalho, foram integrados diversos BD, onde foram considerados como *Concepts* por exemplo: o DrugBank (para a obtenção de compostos), UniProt (Receptor), BLAST, OpenBaBel. Essas bases são então combinadas utilizando o workflow descrito na Figura 9.1(a). Essas fontes de dados heterogêneas são então convertidas para o formato Ondex, mapeadas e transformadas por essa plataforma para criar novas relações entre os dados. Na Figura 9.1(b) adaptada do trabalho de [COC10], é mostrado um grafo gerado com a plataforma Ondex a partir dos dados integrados, mostrando as conexões entre os *Concepts*.

No trabalho é descrito um estudo de caso para a busca de novos fármacos para uma proteína específica, mostrando a eficiência da base de dados integrada e como esta pode revelar conhecimentos para a descoberta de novos usos terapêuticos de drogas [COC10].



(a) (b)

Figura 9.1: Figura adaptada de [COC10] que descreve o BD proposto para Triagem Virtual de Compostos. (a) Workflow de integração de dados desenvolvido em [COC10]. (b) Uma parte do grafo gerado pela plataforma Ondex sobre os dados integrados em [COC10].

9.2 A Execução de Docagem Molecular com o Receptor Flexível e Seleção de Conformações

9.2.1 Módulo de Seleção do Conformações do FReDoWS

O primeiro trabalho desenvolvido pelo LABIO com o objetivo de reduzir o número de conformações e acelerar as simulações de docagem com o modelo FFR está descrito em [MAC06, MAC11a] e consiste em um módulo de seleção de conformações que foi inserido no *workflow* FReDoWS. O critério de seleção utilizado em [MAC06, MAC11a] baseou-se na idéia de que, se um resultado de docagem com determinada conformação obteve um bom valor de FEB e RMSD, é possível que esta mesma conformação, ao interagir com um ligante parecido com o primeiro, também irá apresentar bons valores de FEB e RMSD. Para realizar essa seleção de conformações, são executadas as seguintes etapas:

1. é solicitado ao usuário que informe o total de conformações que ele deseja selecionar, um valor de RMSD máximo e uma tabela de resultados exaustivos para o modelo FFR;

2. a tabela de resultados informada é ordenada de forma crescente por FEB;

3. essa tabela ordenada é separada em duas tabelas de acordo com o valor de RMSD máximo informado;

4. se o total de conformações na tabela com valores dentro do limite de RMSD ultrapassar ou for igual ao total de conformações solicitas pelo usuário, a lista de conformações a serem utilizadas na docagem *Seletiva* está pronta para uso, caso contrário, são adicionadas conformações cujo resultado de docagem excede o valor máximo de RMSD indicado.

Neste trabalho foram realizados estudos de caso para verificar a eficiência do critério de seleção desenvolvido. Nos resultados apresentados em [MAC06, MAC11a] esse critério se mostrou eficiente principalmente quando realizadas comparações entre os resultados exaustivos e seletivos para um mesmo ligante, a seleção representou adequadamente o modelo FFR. Dessa forma, segundo Machado et al. [MAC11a], para ligantes de uma mesma classe não é necessária a utilização de todas as conformações do FFR, sendo o conjunto selecionado já eficiente na representação do todo. Esse trabalho foi o passo inicial para todo o trabalho apresentado nesta Tese.

9.2.2 RCS - *Relaxed Complex Scheme* Proposto por Lin et al. [LIN02, LIN03]

O trabalho apresentado em Lin et al. [LIN03] é uma descrição detalhada do trabalho resumido em [LIN02]. Nesses trabalhos [LIN02, LIN03] foi proposta uma abordagem computacional para o tratamento da flexibilidade de receptores: o RCS - *relaxed-complex scheme*. Esse método reconhece que ligantes devem se ligar a conformações do receptor que ocorrem raramente em uma dinâmica. No RCS, num primeiro estágio é executada uma simulação por DM do receptor sem estar com nenhum ligante, o que gera uma amostra de conformações do mesmo. No segundo estágio do RCS, é executada uma docagem rápida de mini bibliotecas de candidatos a inibidores a um grande conjunto de conformações do receptor geradas no estágio 1 do processo. Para executar a DM foi utilizado o programa AMBER [PEA95] e a duração da simulação foi de 2 ns. A docagem foi executada utilizando o programa AutoDock3.0.5 [MOR98] com o algoritmo LGA.

Um procedimento automático foi desenvolvido tanto para preparar os arquivos para o AutoDock quanto para executar a docagem molecular [LIN02] . E quando esse procedimento é utilizado em conjunto com as simulações pela DM permite a acomodação direta da flexibilidade do receptor. No estudo de caso apresentado foram utilizados as conformações de 10 em 10 ps da DM de 2 ns. A distribuição da FEB variou em torno de 3 *kcal/mol*, indicando a sensibilidade dos resultados de docagem para as diferentes conformações da DM [LIN02].

Segundo Lin et al [LIN03] existem inúmeros métodos que podem ser aplicados para o cálculo da energia livre de ligação (FEB) do complexo, mas a maiorias destes necessitam de um grande esforço computacional. O esquema aplicado em RCS, o MM/PBSA é um método de pós-processamento para avaliar as energias livres de moléculas ou de complexos com um custo computacional mais baixo que outros métodos. Esse esquema combina energias mecânicas moleculares (MM), energia de solvatação e entropia estimada do complexo e estão detalhadas em [LIN03]. Utilizando o esquema MM/PBSA (*Molecular-Mechanics Poisson-Botzmann Surface Area*) e ranqueando por FEB os diferentes modos de ligação receptor-ligante dos complexos minimizados, é demonstrado por [LIN03] que os melhores resultados estão de acordo com a estrutura cristalográfica, o que mostra que com esse esquema MM/PBSA para determinação da FEB foram obtidos complexos mais estáveis.

9.2.3 *Improved* RCS Proposto por Amaro et al. [AMA08]

A Figura 9.2 adaptada de [AMA08] mostra uma revisão do método RCS e indica (processos com o fundo em cinza) onde estão as melhorias apresentadas por Amaro et al. [AMA08].

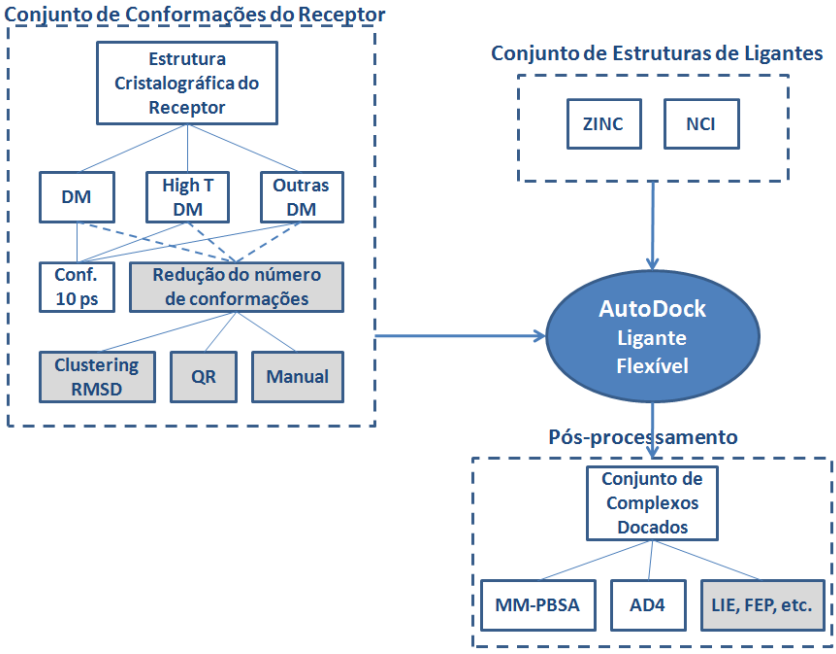


Figura 9.2: Figura adaptada de [AMA08] que mostra uma revisão do método RCS, indicando em fundo cinza as melhorias incluídas por [AMA08] no método RCS. O conjunto de conformações do receptor pode ser gerado por uma DM, ou uma DM com temperatura alta (High T DM), entre outras formas. No RCS de [LIN02, LIN03] foram utilizadas as estruturas de 10 em 10 ps. No trabalho de [AMA08] é proposto 2 algoritmos para seleção de conformações. Também são propostas novas metodologias para o cálculo da FEB: LIE, FEP, explicadas no texto.

A primeira melhora incluída no RCS por Amaro et al. [AMA08] consiste no uso do AutoDock versão 4.0 [MOR09]. Nesta versão do programa AutoDock foram incluídas melhorias em relação a como o ligante é considerado, onde por exemplo, um maior número de tipos de átomos podem estar presentes nos ligantes, é possível adicionar as cargas aos ligantes durante seu preparo para a execução da docagem molecular, entre outras.

O segundo conjunto de melhorias propostas por [AMA08] consiste na consideração tanto de efeitos locais, quanto de efeitos globais na determinação da FEB de determinado complexo receptor-ligante. Essa melhoria esta relacionada com a inclusão de novas funções para a determinação da FEB final, em um estágio de pós-processamento da docagem.

O terceiro conjunto de melhorias define dois algoritmos para a redução do número de conformações da trajetória da DM: a utilização do método Fatoração QR e de um algoritmo de agrupamento de conformações disponíveis no GROMACS [CHR05], que se utiliza dos valores de RMSD entre as estruturas. Essa terceira melhoria proposta por [AMA08] ao método RCS é a que mais interessa nesta revisão pois, assim como o trabalho descrito nesta Tese, foi apresentada uma metodologia para a redução do número de estruturas da DM.

No RCS original as simulações de docagem molecular foram executadas utilizando conformações extraídas da DM em intervalos de tempo iguais, de 10 em 10 ps. O problema é que atualmente as DM são executadas para intervalos de tempo maiores, e por esse motivo, o número de conformações a serem utilizadas aumentou muito. A primeira técnica, chamada de Fatoração QR, foi desenvolvida inicialmente para outros propósitos, e neste trabalho de Amaro et al. [AMA08] foi utilizada para a redução do conjunto de conformações da DM a serem utilizadas na docagem (Figura 9.3. Para isto, essa técnica segue os seguintes passos:

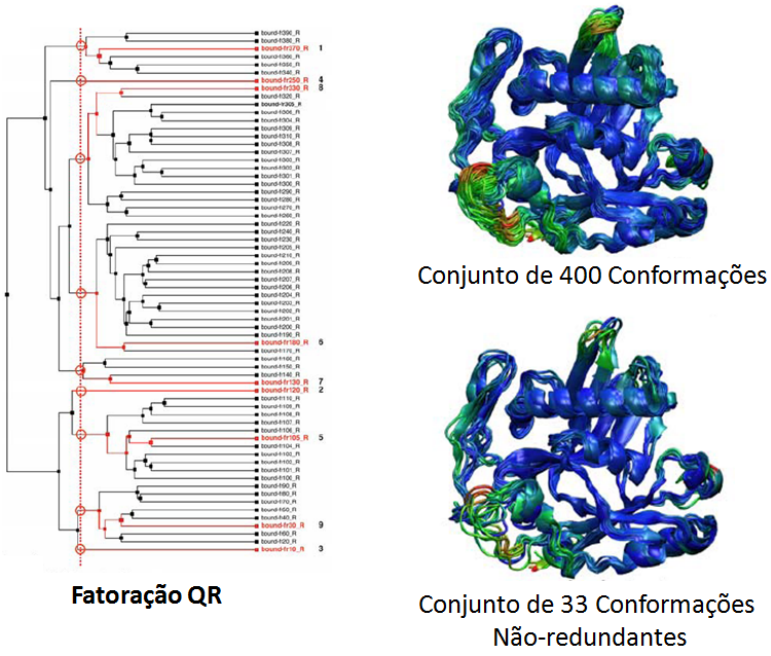


Figura 9.3: Figura adaptada de [AMA08] que mostra o método Fatoração QR incluído no RCS para a seleção da conformações da DM. A esquerda, o resultado da Fatoração QR, que determina as distâncias entre todos os pares de proteínas, de acordo com o RMSD, reordenando as mesmas baseada na similaridade entre elas, onde as não-redundantes estão destacadas em vermelho. A direita, o conjunto inicial de 400 estruturas e o conjunto de 33 estruturas obtidas após a aplicação do método QR.

1. executa um alinhamento estrutural múltiplo utilizando as estruturas da DM de 50 em 50 ps. Para a determinação desse alinhamento múltiplo, todos os possíveis alinhamentos par-a-par são determinados;

2. é realizada um análise a partir de um agrupamento hierárquico, que é computado baseado em uma medida de similaridade estrutural utilizada para o alinhamento múltiplo. A medida de similaridade aplicada mede a distância entre todos os pares de Carbonos α ao longo das estruturas alinhadas;

3. o alinhamento estrutural é então armazenado em uma matriz multidimensional, onde cada conformação é uma coluna e cada linha corresponde a um alinhamento;

4. é aplicado à matriz gerada no passo anterior o método Fatoração QR, gerando como resultado uma lista reordenada das conformações baseada na similaridade entre as mesmas. Esse agrupamento hierárquico reordenado é mostrado à esquerda na Figura 9.3. Ele permite então a definição de um conjunto de conformações não-redundantes de acordo com algum limiar estabelecido pelo usuário (conformações marcadas em vermelho na Figura 9.3). A direita na Figura 9.3, um exemplo da aplicação do método de Fatoração QR, de um conjunto inicial de 400 estruturas foi obtido um conjunto de 33 estruturas não redundantes. Detalhes em [AMA08].

Também no trabalho de [AMA08] é apresentado um método alternativo para o agrupamento de estruturas baseado em uma matriz de RMSD par-a-par das estruturas. Essa matriz de valores de RMSD é dividida em lotes de estruturas mais similares utilizando um algoritmo de agrupamento contido no programa GROMACS [CHR05]. Esse agrupamento permitiu que a docagem fosse executada em um reduzido número de estruturas consideradas mais significativas.

9.3 Agrupamento de trajetórias de DM

9.3.1 Proposta Original [TOR94]

No trabalho de Torda et al. [TOR94] 2 algoritmos de agrupamento tradicionais foram aplicados a uma trajetória pela DM e os mesmos foram comparados com 2 conjuntos de dados de teste. Em [TOR94], primeiro os autores escolheram um subcojunto de todos os átomos pesados da cadeia principal de 12 resíduos de regiões bem conhecidas do receptor para representar as conformações. Após, foi selecionado todos os átomos pesados da cadeia principal dos 64 resíduos do receptor de estudo. Foram aplicados 2 algoritmos de agrupamento: um hierárquico aglomerativo, o *single linkage*, e um hierárquico divisivo.

Independente do algoritmo de agrupamento, o mesmo precisa de uma medida de similaridade para diferenciar os pontos uns dos outros. Nos algoritmos utilizados em [TOR94] a medida considerada foi a de RMSD das coordenadas cartesianas dos átomos considerados, sendo definida por D_{ab} e descrita na Equação 8.1 do Capítulo 8.

Nesse trabalho foram considerados 64 resíduos (ou 12 no primeiro conjunto de teste) de cada uma das 2.000 conformações (estas foram obtidas de uma DM de 1 ns em intervalos de 0.5 ps). Com base nos resultados obtidos, os autores concluem que o algoritmo hierárquico divisivo parece produzir resultados mais significantes do que o algoritmo *single linkage*. Segundo os autores, o *single linkage* falha porque é baseado na distância mínima entre os pontos. Em relação aos resultados com os 2 conjuntos de entradas (com 12 e 64 resíduos) os autores optam pelo conjunto menor, de 12 resíduos. Essa escolha foi feita pois ao considerar todos os resíduos a diferença entre qualquer par de estruturas raramente refletia propriedades conformacionais individuais das mesmas. Em vez disso, esse conjunto refletia mudanças em muitas regiões simultâneas e parcialmente independentes. É importante salientar que neste trabalho não havia o objetivo de utilizar as estruturas agrupadas em simulações de docagem molecular.

9.3.2 Caracterização de Diferentes Algoritmos de Agrupamento [SHA07]

O artigo de Shao et al. [SHA07] é um dos trabalhos mais relacionados com o trabalho apresentado nesta Tese, principalmente em relação ao Capítulo 8. Nesse artigo os autores apresentam um conjunto de 11 algoritmos de agrupamento de diferentes tipos: *Average-Linkage, Single-Linkage, Complete-Linkage, Linkage, Centripetal, Centripetal-Complete, Hierarchical, K-Means, Bayesian, SOM, COBWEB* (10 estão descritos na Seção 4.3.3 do Capítulo 4) implementados e comparados com dados de diferentes simulações por DM. Esses algoritmos foram implementados na linguagem C e foram incorporados ao módulo Ptraj do AMBER9. Todos os 11 algoritmos foram executados com a mesma função de similaridade, a RMSD descrita em [TOR94] e reproduzida pela equação 8.1 do Capítulo 8.

Os testes iniciais apresentados em [SHA07] foram realizados no plano 2D com dados aleatórios, somente para análise dos algoritmos. A seguir os algoritmos foram utilizados considerando como entrada duas trajetórias de 500 ps, em que de cada uma foram extraídos conformações de 5 em 5 ps, totalizando 100 conformações a serem agrupadas. Desses resultados os autores concluíram que o número ideal de grupos é 5. Os autores também realizam experimentos com uma trajetória do mesmo receptor porém agora de 36 ns, em que as conformações foram consideradas de 10 em 10 ps, totalizando 3.644 estruturas a serem agrupadas. Para essa trajetória os testes para definir o número de grupos foi de 2 a 20 grupos onde foram considerados somente alguns resíduos específicos do receptor (um total de 12 resíduos). Dos resultados obtidos com esse último e mais completo experimento os autores concluem :

- a performance dos algoritmos é altamente dependente da escolha do número de grupos e dos átomos utilizados na entrada;
- o algoritmo *single-linkage* é o mais frágil a presença de outliers. Embora esse algoritmo consiga lidar com grupos de diferentes tamanhos, geralmente gera resultados ruins quando os pontos são muito próximos;
- os algoritmos *complete-linkage* e *centripetal* complete são algoritmos hierárquicos aglomerativos que não apresentam grupos com somente um objeto;
- o *centripetal* apresenta resultados similares ao *linkage*, onde apesar de produzirem ótimos valores de DBI, tem muitos grupos com somente um ponto;
- o *linkage* e *average-linkage* apresentam bons resultados para as métricas DBI e pSF. Eles produzem grupos com tamanhos variados;
- o algoritmo *K-means* tende a produzir grupos de tamanhos similares;
- o *Bayesian* produz bons resultados, mas que começam a piorar a medida que o número de grupos aumenta. Para produzir bons resultados, ele deve ser executado muitas vezes, o que gera um alto custo computacional;

- o *SOM* produz também bons resultados porém apresenta dificuldade em produzir grupos de diferentes tamanhos;
- *COBWEB* apesar de um algoritmo promissor também necessita de múltiplas execuções para a obtenção de bons resultados;
- o algoritmo *Hierarchical* foi o mais rápido, sendo muito sensível a *outliers*;
- resumindo, de maneira geral, os autores apontam os algoritmos *K-means*, *average-linkage* e *SOM* com os de melhor performance durante os experimentos.

9.4 Considerações Finais

A base de dados apresentada na primeira seção foi o único trabalho relacionado ao FReDD que foi encontrado até o momento. Apesar do trabalho de [COC10] consistir em uma plataforma de integração de bases de dados públicas, o seu propósito, é o mesmo do FReDD, de auxiliar na descoberta de novos fármacos. A principal diferença é que o FReDD tem um modelo e dados próprios armazenados, enquanto que no trabalho de [COC10] todos os dados provém de bases de dados de acesso público. Além do mais, o [COC10] auxilia o RDD de forma mais direta que o FReDD pois ele já indica possíveis candidatos à fármacos, enquanto que o FReDD será utilizado no futuro com esse propósito.

O módulo de seleção de conformações do workflow FReDoWS [MAC06, MAC11a] se mostrou eficiente para a utilização com conjuntos de ligantes de mesma classe. Porém, um dos objetivos dos trabalhos desenvolvidos no Laboratório de Bioinformática, Modelagem e Simulação de Biossistemas (LABIO) é no futuro realizar Triagem Virtual (do inglês, *Virtual Screening* - VS) com o receptor flexível. Dessa forma, a proposta de seleção de conformações apresentadas em [MAC06, MAC11a] não serviria para esse propósito. Além do mais, neste trabalho não foi realizado nenhuma investigação para o entendimento da interação receptor-ligante, o que está incluído no trabalho descrito nesta Tese. Outro diferencial é que nesta Tese a seleção de conformações foi realizada aplicando-se diferentes técnicas de mineração de dados, incluindo informações do contexto. Esse tipo de análise não está no trabalho de [MAC06, MAC11a].

Nos trabalhos descritos por [LIN02, LIN03] é proposta uma abordagem computacional para o tratamento da flexibilidade de receptores: o RCS. No RCS, é executada uma simulação por DM do receptor e a seguir é executada uma docagem molecular de mini bibliotecas de candidatos a inibidores a um grande conjunto de conformações do receptor geradas na DM. A diferença desta metodologia de execução de docagem com receptor flexível para a empregada no nosso trabalho está no pós-processamento executado no RCS, que utilizando o esquema MM/PBSA, os diferentes modos de ligação receptor-ligante obtidos com a docagem são novamente ranqueados.

Outra diferença significativa do trabalho de [LIN02, LIN03] para o descrito nesta Tese é que, pelo menos uma vez, executamos as simulações de docagem molecular utilizando todas as conformações da DM. Isto permitiu um mapeamento detalhado da interação receptor-ligante com o receptor

flexível. No trabalho de [LIN02, LIN03], as estruturas são utilizadas de 10 em 10 ps, não havendo nenhum tipo de análise nas estruturas entre esses intervalos. O mesmo ocorre no trabalho de [AMA08]. A redução do número de conformações a serem utilizadas é feito com base em estruturas obtidas de 50 em 50 ps da DM. As estruturas entre esses intervalos são ignoradas e não foram analisadas no trabalho. Nós acreditamos que, não realizar nenhum tipo de análise pelo menos uma vez de todas as estruturas pode ocasionar na perda de informações importantes, e principalmente, podem não ser analisadas estruturas que poderiam ter uma melhor afinidade com determinado ligante.

Também no trabalho de [AMA08], apesar dos métodos de agrupamento de conformações serem interessantes, não é demonstrado no trabalho detalhes sobre os mesmos. Além disso, não é apresentada na conclusão do trabalho, qual das duas técnicas de seleção de conformações se mostrou mais efetiva e causou a menor perda de informações.

Tanto para o trabalho de [LIN02, LIN03], quanto para o trabalho de [AMA08], a etapa de pós-processamento aplicada aos resultados de docagem não está disponível para utilização. Sendo assim, não foi possível uma comparação dos resultados desta Tese com os resultados com o método RCS.

Em relação a esses trabalhos de [LIN02, LIN03, AMA08], a nossa grande diferença está no estudo detalhado das interações receptor-ligante. Também apresentamos a utilização de um conjunto de 10 algoritmos de agrupamento e não somente um, conforme descrito em [AMA08]. Além disso, em [AMA08] é sempre utilizado como parâmetro para agrupamento de estruturas o valor do RMSD. No nosso trabalho, modificamos a função de similaridade para também incluir informações do contexto.

O trabalho de Torda et al. [TOR94] é um dos primeiros trabalhos relacionados ao agrupamento de conformações de trajetórias de DM. A principal diferença deste trabalho para o trabalho descrito nesta Tese (em especial aos resultados descritos no Capítulo 8) é que neste somente foram estudados 2 algoritmos de agrupamento, ambos com a função de similaridade RMSD. Além do mais, neste trabalho os autores não tinham o objetivo de utilizar as estruturas agrupadas para simulações de docagem molecular. Uma característica importante do trabalho de [TOR94] que foi utilizado nesta Tese consiste na análise dos agrupamentos utilizando como entrada diferentes conjuntos de resíduos do receptor. A diferença é que para nossos resultados não foram encontradas diferenças significativas para as diferentes entradas analisadas, enquanto que em [TOR94] os autores indicam ser melhor utilizar conjuntos menores de resíduos a todos os resíduos do receptor de estudo.

O trabalho de Shao et al. [SHA07], apesar de ter sido utilizado como base no desenvolvimento dos experimentos de agrupamento desta Tese, há diferenças significativas entre os trabalhos. A principal é que nesta Tese foram analisadas diferentes funções de similaridade para calcular a distância entre as conformações que estão sendo agrupadas. Essas diferentes funções consideram informações do contexto para melhorar os agrupamentos. Além do mais, no presente trabalho são comparados os resultados ao se considerar diferentes átomos na entrada dos algoritmos, enquanto que em [SHA07] para os experimentos mais importantes foram todos realizados considerando somente alguns resíduos definidos pelos autores como mais importantes. Além do mais, em [SHA07] não há a intenção de utilizar os agrupamentos para a docagem molecular.

10. Considerações Finais

Este documento apresentou todas as etapas do trabalho desenvolvido com os objetivos de melhorar o entendimento sobre a importância da flexibilidade de receptores em docagem molecular e de selecionar conformações do receptor de forma a acelerar esse processo. Como método para alcançar esses objetivos aplicou-se um processo de KDD, em que diferentes técnicas de mineração de dados foram utilizadas. A maioria dos resultados obtidos nesta Tese já está publicada em artigos, resumos, capítulo de livro ou estão em artigos sob revisão: [MAC07, MAC08b, MAC08a, WIN09, WIN10a, WIN10b, COH10, COH11, MAC11a, MAC10c, MAC10b, MAC10d, WIN10c, MAC11b, MAC10a, WIN11].

O Capítulo 2 descreve o embasamento teórico necessário para entendimento desta Tese. Neste capítulo é descrito o estado da arte sobre as principais abordagens utilizadas para a incorporação da flexibilidade de receptores em docagem molecular. É demonstrado alguns exemplos de trabalhos anteriores que indicam que utilizar um conjunto de conformações do receptor, executando uma série de simulações de docagem, é um abordagem interessante e capaz de indicar informações sobre a interação de complexo receptor-ligante impossíveis de serem obtidas de uma docagem com receptor rígido.

Os Capítulos 3 e 4 apresentam os materiais e métodos utilizados para o desenvolvimento deste trabalho. Ao final deste capítulo é descrito o primeiro trabalho que originou todos os resultados posteriores [MAC08b, MAC08a]. Neste foi desenvolvido um BD inicial para armazenamento dos resultados de docagem e das conformações da DM, e a partir desses dados, foram executados os primeiros experimentos de mineração de dados com a técnica de Associação. Como esse modelo de BD não suportava diferentes simulações de docagem, este foi evoluído para o modelo descrito no capítulo seguinte.

Assim, no Capítulo 5, é descrito o primeiro resultado desta Tese, o BD FReDD [WIN09, WIN10a, WIN10b], que armazena os resultados de conformações do receptor e do ligante e de docagem molecular. A partir dos dados armazenados no FReDD, uma etapa de preparação para a mineração foi realizada, onde foi utilizado principalmente as distâncias entre os resíduos do receptor e os 4 ligantes estudados. Ao final deste capítulo é realizada uma análise preliminar nos resultados armazenados no FReDD que selecionam um conjunto de 25 resíduos do receptor que mais interagem com os 4 ligantes. Esses resíduos, chamados de Top 25, são utilizados no Capítulo 8 para as análises com a técnica de agrupamento.

O Capítulo 6 apresenta o segundo conjunto de resultados desta Tese, a aplicação da técnica de mineração de dados Classificação com árvores de decisão utilizando o algoritmo J48 [MAC10c, MAC10b, MAC11b, WIN10b]. Uma das principais contribuições dessa capítulo é a metodologia proposta de discretização do atributo-alvo dos arquivos de entrada utilizados. Essa metodologia proposta é comparada com 2 métodos de discretização clássicos com base no impacto dos mesmos no resultado das árvores de decisão obtidas. Os resultados com a Classificação apesar de gerar modelos interessantes e permitir que fossem extraídos conhecimentos sobre a interação receptor-

ligante, a utilização para a seleção de conformações do receptor em docagem com ligantes diferentes não é possível de ser feita diretamente pois as conformações do receptor com melhor FEB são diferentes para os 4 ligantes, não sendo possível selecionar um conjunto único de conformações mais promissoras. Além do mais, a discretização não é precisa uma vez que a variação dos valores de FEB entre as instâncias de entrada é muito sutil, prejudicando a determinação de que uma instância pertencia a uma classe ou a outra. Assim, optou-se pelo uso de um algoritmo onde não fosse necessária a discretização do atributo-classe FEB: o algoritmo escolhido foi o de regressão M5P.

Os resultados com a aplicação da técnica de mineração de Regressão com o algoritmo de árvores modelo M5P são resumidos no Capítulo 7 [MAC10d, WIN10c, MAC11b, MAC10a, WIN11]. As principais contribuições deste capítulo estão relacionadas ao pré-processamento dos dados baseado no contexto e a metodologia de pós processamento dos resultados das árvores modelo que permitiu a indicação das conformações mais promissoras nesses experimentos. Apesar dos resultados com o M5P serem interessantes, assim como para Classificação, a utilização dos mesmos, diretamente para seleção de conformações em futuras simulações de docagem molecular não é promissora. O principal problema encontrado é que as melhores conformações são diferentes para cada ligante. Ou seja, não é possível, a partir desses resultados, estabelecer um conjunto único de conformações mais relevantes. Outro problema encontrado é que, para se utilizar os modelos induzidos para predizer o valor de FEB de novos ligantes é necessário saber as distâncias mínimas dos mesmos para os resíduos do receptor, informação que somente é obtida após a execução da docagem molecular, o que também dificulta a utilização dos modelos com o M5P para efetivamente selecionar conformações do receptor para compostos ainda não testados. Por esses motivos, optou-se por não mais se utilizar como entrada nos experimentos de mineração os resultados de docagem molecular e sim, diretamente as conformações do receptor. E, como não será mais utilizado os resultados de docagem, não tem-se mais um atributo-classe FEB. A técnica de aprendizado não-supervisionado escolhida foi a de Agrupamento.

O Capítulo 8 apresenta o último conjunto de resultados desta Tese, que compreende os experimentos com a técnica de Agrupamento. Neste capítulo são descritos uma série de experimentos executados com diferentes configurações, incluindo a descrição de cinco novas funções de similaridade desenvolvidas com o objetivo de melhorar os agrupamentos considerando informações sobre o contexto dos dados. No final deste capítulo são descritas análises com o P-MIA [HÜB10], que comparam as funções de similaridade mostrando um estudo de caso efetivo do ganho de processamento obtido com a utilização do P-MIA em conjunto com os resultados de Agrupamento. Apesar da análise com o P-MIA ter sido realizada com somente uma das configurações de experimento de agrupamento, está já mostra um ganho de processamento interessante, tanto utilizando a função RMS padrão implementada em [SHA07] quanto com a função proposta nesta Tese, a TCN_Mult2. Neste estudo de caso, com somente 20% das conformações processadas, houve ganhos de aproximadamente 50% (50% das conformações foram descartadas) o que possibilita a execução dos experimentos de docagem em um tempo consideravelmente mais reduzido. Além do mais, com os mesmos 20% de processamento, 77% (RMS) e 82% (TCN_Mult2) das melhores conformações

foram consideradas. Ou seja, com 20% do tempo de um experimento exaustivo, 80% das melhores conformações já foram consideradas.

O Capítulo 9 relaciona alguns trabalhos já publicados com o conteúdo desta Tese, que incluem trabalhos sobre BD para Desenho Racional de Fármacos, trabalhos sobre a execução de docagem molecular com o receptor flexível e seleção de conformações e trabalhos sobre a utilização de algoritmos de agrupamento com dados de DM. A discussão ao final deste capítulo mostra que apesar dos trabalhos relacionados abordarem a flexibilidade do receptor da mesma forma que a apresentada neste trabalho, o estudo detalhado da importância da flexibilidade e a aplicação de um processo de KDD nesse tipo de resultado de docagem molecular com o FFR são os diferenciais deste trabalho.

Com base em todos os resultados apresentados, desde o BD FReDD até os experimentos com Classificação, Regressão e Agrupamento, este trabalho contribuiu para melhorar a eficiência da seleção de conformações do receptor utilizando um processo completo de KDD, uma vez que os dados foram preparados, a mineração de dados foi aplicada e os resultados foram pós-processados. Com a implementação do P-MIA e com as novas DM que estão sendo executadas no LABIO, possivelmente todas as contribuições deste trabalho serão efetivamente utilizadas para a busca de novos compostos para a InhA e para outros receptores que venham a ser alvo de estudo no laboratório.

10.1 Principais Contribuições

As principais contribuições obtidas com o desenvolvimento desta Tese atendem ao principal objetivo da mesma: *contribuir para o entendimento da importância da flexibilidade do receptor em simulações de docagem molecular e para a redução do tempo necessário para a execução desse tipo de experimento a partir da aplicação de um processo de descoberta de conhecimento em Banco de Dados*:

- O modelo do BD FReDD desenvolvido para armazenamento de resultados de docagem com o FFR e de conformações resultantes de DM. Não foi encontrado outro BD que apresentasse um modelo para este mesmo tipo de dado biológico.

- O algoritmo desenvolvido para a preparação dos dados de docagem com o FFR para utilização nas técnicas de mineração de dados. A utilização de distâncias mínimas entre os resíduos do receptor e os ligantes como atributos preditivos, obtidas a partir dos dados armazenados no FReDD, é uma contribuição interessante desta Tese pois pode ser aplicada como uma nova forma de análise dos resultados de interação receptor-ligante (como a análise apresentada ao final do Capítulo 5).

- A aplicação de técnicas de mineração de dados em resultados de docagem molecular com o receptor flexível para a extração de conhecimento sobre a interação do complexo receptor-ligante.

- O método de discretização proposto, que utiliza os valores de Média e Desvio Padrão do atributo-alvo para a determinação das classes. Para o atributo-alvo FEB, esse método de discretização foi o mais promissor a gerou as melhores árvores de decisão.
- A seleção de atributos baseada no contexto, que nos experimentos com o algoritmo M5P de Regressão melhoraram os modelos gerados.
- O método de pós-processamento das árvores modelo geradas com o M5P que permitiram que fossem selecionados conjuntos de conformações mais promissoras para cada ligante.
- As novas funções de similaridade para os algoritmos de Agrupamento: TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ e a comparação das mesmas em relação aos resultados obtidos para métricas clássicas como DBI e pSF e em relação a aplicação, considerando as médias de DP de FEB dos agrupamentos para diferentes configurações de experimentos.
- Os agrupamentos gerados (independente da função) que, utilizados em conjunto com o P-MIA, permitem um ganho no processamento de experimentos de docagem com o modelo FFR do receptor.

10.2 Trabalhos Futuros

Como sugestões para trabalhos futuros:

- Executar os experimentos de mineração de dados a partir dos resultados de DM mais longas. Atualmente, estão sendo produzidas no LABIO DM de 10 até 100 ns, o que está gerando um número muito maior de conformações a serem consideradas em docagem molecular com o modelo FFR.
- Expandir o BD FReDD, disponibilizando acesso ao mesmo pela Web, para que outros grupos de pesquisa tenham acesso aos dados armazenados.
- Com a implementação do P-MIA, que está em fase de execução, analisar outras configurações de agrupamentos e seu impactos no ganho de processamento, o que também permitirá que novas funções de similaridade sejam desenvolvidas e testadas.
- Paralelizar a execução dos experimentos de agrupamento, pois para a DM analisada, de 3,1 ns, por exemplo o algoritmo K-means despende em torno de 2 horas para executar, considerando somente uma determinada configuração de experimento. Para a utilização de Agrupamento nas DM que estão sendo executadas no LABIO, será necessária essa paralelização dos algoritmos.
- Executar outras técnicas de mineração de dados diretamente com as coordenadas cartesianas das conformações, como o trabalho que está sendo desenvolvido pela doutoranda Ana Winck.

REFERÊNCIAS

[ADM10] C.P. Adams, V.V. Brantner. "Spending on new drug development". *Health Economics*, vol. 19, 2010, pp. 130–141.

[AGR93] R. Agrawal, T. Imielinski, A. Swami. "Mining association rules between sets of items in large databases". In: ACM-SIGMOD Int. Conf. Management of Data (SIGMOD93), 1993, pp. 207–216.

[AGU08] F. Aguero, B. Al-Lazikani, M. Aslett, M. Berriman, F.S. Buckner, R.K. Campbell, S. Carmona, I.M. Carruthers, A.W. Chan, F. Chen, G.J. Crowther, M.A. Doyle, C. Hertz-Fowler, A.L. Hopkins, G. McAllister, S. Nwaka, J.P. Overington, A. Pain, G.V. Paolini, U. Pieper, S.A. Ralph, A. Riechers, D.S. Roos, A. Sali, D. Shanmugam, T. Suzuki, W.C. Van Voorhis, C.L. Verlinde. "Genomic-scale prioritization of drug targets: the TDR targets database". *Nat. Rev. Drug Discov.*, vol. 7, 2008, pp. 900–907.

[ALO06] H. Alonso, A.A. Bliznyuk, J.E. Gready. "Combining docking, molecular dynamic simulations in drug design". *Med. Res. Rev.*, vol. 26, 2006, pp. 531–568.

[AMA08] R.E. Amaro, R. Baron, J.A. McCammon. "An improved relaxed complex scheme for receptor flexibility in computer-aided drug design". *J. Comput. Aided Mol. Des.*, vol. 22, 2008, pp. 693–705.

[APO98] J. Apostolakis, A. Plückthun, A. Caflisch. "Docking small ligands in flexible binding sites". *J. Comput. Chem.*, vol. 19, 1998, pp. 21–37.

[BAU00] A.R. Baulard, J.C. Betts, J. Engohang-Ndong, S. Quan, R.A. McAdam, P.J. Brennan, C. Locht, G.S. Besra. "Activation of the pro-drug ethionamide is regulated in mycobacteria". *J. Biol. Chem.*, vol. 275, 2000, pp. 28326–28331.

[BER00] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. "PDB - Protein Data Bank". *Nucl. Acids Res.*, vol. 28, 2000, pp. 235–242.

[BOT09] G. Bottegoni, I. Kufareva, M. Totrov, R. Abagyan. "Four-dimensional docking: a fast and accurate account of discrete receptor flexibility in ligand docking". *J. Med. Chem.*, vol. 52, 2009, pp. 397–406.

[BRA09] C. B-Rao, J. Subramanian, S.D. Sharma. "Managing protein flexibility in docking and its applications". *Drug Discov. Today*, vol. 14, 2009, pp. 394–398.

[BRO00] H.B. Broughton. "A method for including protein flexibility in protein-ligand docking: improving tools for database mining and virtual screening". *J. Mol. Graph. Model*, vol. 18, 2000, pp. 247–257.

[CAR00] H.A. Carlson, J.A. McCammon. "Accommodating protein flexibility in computational drug design". *Mol. Pharmacol.*, vol. 57, 2000, pp. 213–218.

[CAS99] D.A. Case, D.A. Pearlman, J.W. Caldwell, T.E. Cheatham III, W.R. Ross, C.L. Simmerling, T.A. Darden, K.M. Merz, R.V. Stanton, A.L. Cheng, J.J. Vincent, M. Crowley, V. Tsui, R.J. Radmer, Y. Duan, J. Pitera, I. Massova, G.L. Seibel, U.C. Singh, P.K. Weiner, P.A. Kollman. "AMBER 6". Manual do Usuário. University of California, San Francisco, 1999, 422p.

[CAS06] D.A. Case, T.A. Darden, T.E. Cheatham III, C.L. Simmerling, J. Wang, R.E. Duke, R. Luo, K.M. Merz, D.A. Pearlman, M. Crowley, R. C. Walker, W. Zhang, B. Wang, S. Hayik, A. Roitberg, G. Seabra, K. F. Wong, F. Paesani, X. Wu, S. Brozell, V. Tsui, H. Gohlke, L. Yang, C. Tan, J. Mongan, V. Hornak, G. Cui, P. Beroza, D.H. Mathews, C. Schafmeister, W. S. Ross, P.A. Kollman. "AMBER 9". Manual do Usuário. University of California, San Francisco, 2006, 328p.

[CAS07] C.T. Caskey. "The drug development crisis: Efficiency and safety". *Annu. Rev. Med.*, vol. 58, 2007, pp. 1–16.

[CER09] N.M. Cerqueira, N.F. Bras, P.A. Fernandes, M.J. Ramos. "Madamm: A multistaged docking with an automated molecular modeling protocol". *Proteins*, vol. 74, 2009, pp. 192–206.

[CHE96] M-S. Chen, J. Han, P.S. Yu. "Data mining: An overview from database perspective". *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, 1996, pp. 866–883.

[CHR05] M. Christen, P. H. Hunenberger, D. Bakowies, R. Baron, R. Burgi, D.P. Geerke, T.N. Heinz, M.A. Kastenholz, V. Krautler, C. Oostenbrink, C. Peter, D. Trzesniak, W.F. van Gunsteren. "The gromos software for biomolecular simulation: Gromos05". *J. Comput. Chem.*, vol. 26, 2005, pp. 1719–1751.

[CLA01] H. Claussen, C. Buning, M. Rarey, T. Lengauer. "FlexE: Efficient molecular docking considering protein structure variations". *J. Mol. Biol.*, vol. 308, 2000, pp. 377–395.

[COC09] P.J.A. Cock, T. Antao, J.T. Chang, B.A. Chapman, C.J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, M.J.L. de Hoon. "BioPython: Freely available python tools for computational molecular biology and bioinformatics". *Bioinformatics*, vol. 25, 2009, pp. 1422–1423.

[COC10] S.J. Cockell, Jochen W., P. Lord, C. Wipat, D. Andriychenko, M. Pocock, D. Wilkinson, M. Young, A. Wipat. "An integrated dataset for in silico drug discovery". *Journal of Integrative Bioinformatics*, vol. 7, 2010, pp. 116–129.

[COH09] E.M.L. Cohen. "Um estudo do efeito da flexibilidade explícita da enzima InhA de *M. tuberculosis* na docagem molecular dos inibidores etionamida, triclosano e isoniazida-pentacionoferrato II". Dissertação de Mestrado, Programa de Pós-graduação em Biologia Celular e Molecular, PUCRS, Porto Alegre, RS, Brasil, 2009, 59p.

[COH10] E.M.L. Cohen, K.S. Machado, O. Norberto de Souza. "The effect of InhA flexibility in docking simulations with ethionamide and triclosan". In: International Society for Computational Biology Latin America Conference, 2010, pp. 73–73.

[COH11] E.M.L. Cohen, K.S. Machado, M. Cohen, O. Norberto de Souza. "Effect of the explicit flexibility of the InhA enzyme from *Mycobacterium tuberculosis* in molecular docking simulations." *BMC Bioinformatics*, Em revisão, 2011, 25p.

[COZ08] P. Cozzini, G.E. Kellogg, F. Spyrakis, D.J. Abraham, G. Costantino, A. Emerson, F. Fanelli, H. Gohlke, L.A. Kuhn, G.M. Morris, M. Orozco, T.A. Pertinhez, M. Rizzi, C.A. Sotriffer. "Target flexibility: An emerging consideration in drug discovery and design". *J. Med. Chem.*, vol. 51, 2008, pp. 6237–6255.

[DAV79] D.L. Davies, D.W. Bouldin. "A cluster separation measure". *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 1, 1979, pp. 224–227.

[DES95] A. Dessen, A. Quemard, J.S. Blanchard, W.R. Jacobs, J.C. Sacchettini. "Crystal Structure and Function of the Isoniazid Target of *Mycobacterium tuberculosis*". *Science*, vol. 267, 1995, pp. 1638–1641.

[DOU95] J. Dougherty, R. Kohavi, M. Sahami. "Supervised and unsupervised discretization of continuous features". In: 12th International Conference on Machine Learning, 1995, pp. 194–202.

[EWI01] T.J.A. Ewing, S. Makino, A.G. Skillman, I.D. Kuntz. "DOCK4.0: Search Strategies for Automated Molecular Docking of Flexible Molecule Databases". *J. Comp. Aided. Mol. Des.*, vol. 15, 2001, pp. 411–428.

[FAY96] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. "The KDD process for extracting useful knowledge from volumes of data". *Communications of the ACM*, vol. 39, 1996, pp. 27–34.

[FER04] A.M. Ferrari, B.Q. Wei, L. Costantino, B.K. Shoichet. "Soft docking and multiple receptor conformations in virtual screening". *J. Med. Chem.*, vol. 47, 2004, pp. 5076–5084.

[FRE10] A.A. Freitas, D.C. Wieser, R. Apweiler. "On the importance of comprehensible classification models for protein function prediction". *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 99, 2010, pp. 172–182.

[GOO96] D. Goodsell, G. Morris, A. Olson. "Automated Docking of Flexible Ligands: Applications of AutoDock". *J. Mol. Recognit.*, vol. 9, 1996, pp. 1–5.

[GUE97] N. Guex, M.C. Peitsch. "SWISS-MODEL and the Swiss-PDBViewer: An Environment for Comparative Protein Modeling". *Electrophoresis*, vol. 18, 1997, pp. 2714–2723.

[GUH98] S. Guha, R. Rastogi, K. Shim. "CURE: an efficient clustering algorithm for large databases". *SIGMOD Rec.*, vol. 27, 1998, pp. 73–84.

[HAL00] M. Hall. "Correlation-based feature selection for discrete and numeric class machine learning". In: International Conference on Machine Learning, 2000, pp. 359–366.

[HAL09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten. "The WEKA data mining software: an update". *SIGKDD Explor. Newsl.*, vol. 11, 2009, pp. 10–18.

[HAN02] J. Han. "How can data mining help bio-data analysis?". In: 2nd ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD 2002), 2002, pp. 1–2.

[HAN06] J. Han, M. Kamber. "Data Mining: Concepts and Techniques". New York: Morgan Kaufmann, 2006, 2° Edição, 743p.

[HAR79] J.A. Hartigan, M.A. Wong. "A K-means clustering algorithm". *Applied Statistics*, vol. 28, 1979, pp. 100–108.

[HOU99] T. Hou, J. Wang, L. Chen, X. Xu. "Automated docking of peptides and proteins by using a genetic algorithm combined with a tabu search". *Protein Eng.*, vol. 12, 1999, pp. 639–647.

[HUA06] S.Y. Huang, X. Zou. "Ensemble docking of multiple protein structures: Considering protein structural variations in molecular docking". *Proteins*, vol. 66, 2006, pp. 399–421.

[HÜB10] P.N. Hübler. "P-MIA: Padrão Múltiplas Instâncias AutoAdaptáveis- um Padrão de Dados para Workflows Científicos". Tese de Doutorado, Programa de Pós-graduação em Ciência da Computação, PUCRS, Porto Alegre, RS, Brasil, 2010, 179p.

[HUM96] W. Humphrey, A. Dalke, K. Schulten. "VMD - Visual Molecular Dynamics". *J. Mol. Graph.*, vol. 14, 1996, pp. 33–38.

[IRW05] J.J. Irwin, B.K. Shoichet. "ZINC – a free database of commercially available compounds for virtual screening." *J. Chem. Inf. Model.*, vol. 45, 2005, pp. 177–182.

[JEF97] G.A. Jeffrey. "An introduction to hydrogen bonding". New York: Oxford University Press, 1997, 3° Edição, 320p.

[JIA91] F. Jiang, S.H. Kim. "Soft docking: Matching of molecular surface cubes". *J. Mol. Biol.*, vol. 219, 1991, pp. 79–102.

[JOH67] S.C. Johnson. "Hierarchical clustering schemes". *Psychometrika*, vol. 3, 1967, pp. 241–254.

[KAR00] M. Karplus. "Aspects of protein reaction dynamics: Deviations from simple behavior". *J. Phys. Chem.*, vol. 104, 2000, pp. 11–27.

[KNE97] R.M. Knegtel, I.D. Kuntz, C.M. Oshiro. "Molecular docking to ensembles of protein structures". *J. Mol. Biol.*, vol. 266, 1997, pp. 424–440.

[KOH06] J. Kohler, J. Baumbach, J. Taubert, M. Specht, A. Skusa, E. Ruegg, C. Rawlings, P. Verrier. "Graph-based analysis and visualization of experimental results with Ondex". *Bioinformatics*, vol. 22, 2006, pp. 1390–1383.

[KUN92] I.D. Kuntz. "Structure-based strategies for drug design and discovery". *Science*, vol. 257, 1992, pp.1078–1082.

[KUO03] M.R. Kuo, H.R. Morbidoni, D. Alland, S.F. Sneddon, B.B. Gourlie, M.M. Staveski, M. Leonard, J.S. Gregory, A.D. Janjigian, C. Yee, J.M. Musser, B. Kreiswirth, H. Iwamoto, R. Perozzo, W.R. Jacobs, J.C. Sacchettini, D.A. Fodock. "Targeting Tuberculosis and Malaria through Inhibition of Enoyl Reductase: Compound Activity and Structural Data". *J. Biol. Chem.*, vol. 278, 2003, pp. 20851–20859.

[LEA94] A.R. Leach. "Ligand docking to proteins with discrete side-chain flexibility". *J. Mol. Biol.*, vol. 235, 1994, pp. 345–356.

[LEA98] A.R. Leach, A.P. Lemon. "Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm". *Proteins*, vol. 33, 1998, pp. 227–239.

[LEN96] T. Lengauer, M. Rarey. "Computational methods for biomolecular docking". *Curr. Opin. Struct. Biol.*, vol. 6, 1996, pp. 402–406.

[LIN02] J-H. Lin, A.L. Perryman, J.R. Schames, J.A. McCammon. "Computational drug design accommodating receptor flexibility: The relaxed complex scheme". *J. Am. Chem. Soc.*, vol. 124, 2002, pp. 5632–5633.

[LIN03] J-H. Lin, A.L. Perryman, J.R. Schames, J.A. McCammon. "The relaxed complex method: Accommodating receptor flexibility for drug design with an improved scoring scheme". *Biopolymers*, vol. 68, 2003, pp. 47–62.

[LYB95] T.P. Lybrand. "Ligand-protein docking and rational drug design". *Curr. Opin. Struct. Biol.*, vol. 5, 1995, pp. 224–228.

[LYN02] P. Lyne. "Structure-based virtual screening: an overview". *Drug Discov. Today*, vol. 7, 2002, pp. 1047–1055.

[MAC06] K.S. Machado. "Um workflow científico para a modelagem do processo de desenvolvimento de fármacos assistido por computador utilizando receptor flexível". Dissertação de mestrado, Programa de Pós-graduação em Ciência da Computação, PUCRS, Porto Alegre, RS, Brasil, 2006, 77p.

[MAC07] K.S. Machado, E.K. Schroeder, D.D. Ruiz, O. Norberto de Souza. "Automating molecular docking with explicit receptor flexibility using scientific workflows". *LNBI-LNCS Advances in Bioinformatics and Computational Biology*, vol. 4643, 2007, pp. 1–11.

[MAC08a] K.S. Machado, E.K. Schroeder, D.D. Ruiz, A.T. Winck, O. Norberto de Souza. "Extracting information from flexible receptor-flexible ligand docking experiments". *LNBI-LNCS Advances in Bioinformatics and Computational Biology*, vol. 5167, 2008, pp.104–114.

[MAC08b] K.S. Machado, E.K. Schroeder, D.D. Ruiz, O. Norberto de Souza. "Extracting information from flexible ligand-receptor docking experiments". In: 3rd Conference of the Brazilian Association for Bioinformatics and Computational Biology - X-Meeting, 2007, pp. 1–1.

[MAC10a] K.S. Machado, A.T. Winck, D.D. Ruiz, O. Norberto de Souza. "Mining flexible-receptor docking experiments to select promising protein receptor snapshots". *BMC Genomics*, vol. 11, 2010, pp. 1–13.

[MAC10b] K.S. Machado, A.T. Winck, D.D. Ruiz, O. Norberto de Souza. "Comparison of discretization methods of flexible-receptor docking data for analyses by decision trees". In: IADIS International Conference Applied Computing, 2010, pp. 223–229.

[MAC10c] K.S. Machado, A.T. Winck, D.D. Ruiz, O. Norberto de Souza. "Discretization of flexible-receptor docking data". *LNBI-LNCS Advances in Bioinformatics and Computational Biology.*, vol. 6268, 2010, pp. 75–79.

[MAC10d] K.S. Machado, A.T. Winck, D.D. Ruiz, O. Norberto de Souza. "Applying model trees on flexible-receptor docking experiments to select promising protein receptor snapshots". In: International Society for Computational Biology Latin America Conference, 2010, pp. 66–66.

[MAC11a] K.S. Machado, E.K. Schroeder, D.D. Ruiz, E.M.L. Cohen, O. Norberto de Souza. "FReDoWS: A method to automate molecular docking simulations with explicit receptor flexibility and snapshots selection". *BMC Bioinformatics*, Em revisão. 2° rodada, 2011, 20p.

[MAC11b] K.S. Machado, A.T. Winck, D.D. Ruiz, E.M.L. Cohen, O. Norberto de Souza. "Mining flexible-receptor docking data". *WIREs Data Mining and Knowledge Discovery*, Em revisão. 3° rodada, 2011, 13p.

[McD94] I.K. McDonald, J.M. Thornton. "Satisfying hydrogen bonding potential in proteins". *J. Mol. Biol.*, vol. 238, 1994, pp. 777–793.

[MOR98] G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Hart, R.K. Belew, A.J. Olson. "Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function". *J. Comput. Chem.*, vol. 19, 1998, pp. 1639–1662.

[MOR01] G.M. Morris, D.S. Goodsell, R. Huey, W. E. Hart, S. Halliday, R. Belew, A.J. Olson. "AutodDock User's Guide - AutoDock: Automated Docking of Flexible ligands and receptors. Version 3.0.5.", Manual do usuário, Department of Molecular Biology, The Scripps Research Institute, La Jolla, 2001, 86p.

[MOR09] G.M. Morris, R. Huey, W. Lindstrom, M.F. Sanner, R.K. Belew, D.S. Goodsell, A.J. Olson. "AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility". *J. Comput. Chem.*, vol. 30, 2009, pp. 2785–2791.

[OLI04] J.S. Oliveira, E.H.S. Sousa, L.A. Basso, M. Palaci, R. Dietze, D.S. Santos, I. Moreira. "An Inorganic Iron Complex that Inhibits Wild-type and an Isoniazid-resistant Mutant 2-trans-enoyl-ACP (CoA) Reductase from *Mycobacterium tuberculosis*". *Chem. Commun.*, vol. 15, 2004, pp. 312–313.

[OLI07] J.S. Oliveira, I.S. Moreira, D.S. Santos, L.A. Basso. "Enoyl reductases as targets for the development of anti-tubercular and anti-malarial agents". *Current Drug Targets*, vol. 8, 2007, pp. 399–411.

[OST02] F. Osterberg, G.M. Morris, M.F. Sanner, A.J. Olson, D.S. Goodsell. "Automated docking to multiple target structures: incorporation of protein mobility and structural water heterogeneity in AutoDock". *Proteins*, vol. 46, 2002, pp. 34–40.

[ORG08] World Health Organization. "Tuberculosis Facts. WHO 2008", Capturado em: http://www.who.int/tb/publications/2008/en/index.html, Janeiro 2011.

[PEA95] D.A. Pearlman, D. A. Case, J.W. Caldwell, W.R. Ross, T.E. Cheatham, S. DeBolt,D. Ferguson, G. Seibel, P. Kollman. "AMBER, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules". *Comp. Phys. Commun.*, vol. 91, 1995, pp. 1–41.

[POS11] Python Software Foundation. "The Python Tutorial, 2011". Capturado em: http://docs.python.org/py3k/tutorial/index.html, Fevereiro 2011.

[QUI86] J.R. Quinlan. "Induction of decision trees". *Mach. Learn.*, vol. 1, 1986, pp. 81–106.

[QUI92] J.R. Quinlan. "Learning with continuous classes". In: 5th Australian Joint Conference on Artificial Intelligence, 1992, pp. 343–348.

[RAR96] M. Rarey, B. Kramer, T. Lengauer, G. Klebe. "A Fast Flexible Docking Method Using an Incremental Construction Algorithm". *J. Mol. Biol.*, vol. 261, 1996, pp. 470–489.

[ROZ98] D.A. Rozwarski, G.A. Grant, D.H. Barton, W.R. Jacobs Jr., J.C. Sacchettini. "Modification of the NADH of the Isoniazid Target (InhA) from *Mycobacterium tuberculosis*". *M Science*, vol. 279, 1998, pp. 98–102.

[SCH98] L. Schaffer, G.M. Verkhivker. "Predicting structural effects in HIV-1 protease mutant complexes with flexible ligand docking and protein side-chain optimization". *Proteins*, vol. 33, 1998, pp. 295–304.

[SCH00] V. Schnecke, L.A. Kuhn. "Virtual screening with solvation and ligand-induced complementarity". *Perspectives in Drug Discovery and Design*, vol. 20, 2000, pp. 171–190.

[SCH04] E.K. Schroeder. "Análise computacional da Enzima 2-trans-Enoil-ACP(CoA) Redutase de *Mycobacterium tuberculosis*, produto do gene inhA, como alvo para o desnvolvimento de drogas anti-tuberculose". Tese de Doutorado, Programa de Pósgraduação em Biologia Celular e Molecular, UFRGS, Porto Alegre, RS, Brasil, 2004, 264p.

[SCH05] E.K. Schroeder, L.A. Basso, D.S. Santos, O. Norberto de Souza. "Molecular Dynamics Simulation Studies of the Wild-Type, I21V, and I16T Mutants of Isoniazid-Resistant *Mycobacterium tuberculosis* Enoyl Reductase (InhA) in Complex with NADH: Toward the Understanding of NADH-InhA Different Affinities". *Biophys. J.*, vol. 89, 2005, pp. 876–884.

[SHA07] J. Shao, S.W. Tanner, N. Thompson, T.E. Cheatham. "Clustering molecular dynamics trajectories: 1. characterizing the performance of different clustering algorithms". *J. Chem. Theory Comput.*, vol. 3, 2007, pp. 2312–2334.

[SIL04] R.B. Silverman. "The organic chemistry of drug design and drug action". Waltham: Academic Press, 2004, 2° Edição, 617p.

[SIL09] C.H. da Silveira, D.E.V. Pires, R.C. Minardi, C. Ribeiro, C.J.M. Veloso, J.C.D. Lopes, W. Jr. Meira, G. Neshich, C.H.I. Ramos, R. Habesh, M.M. Santoro. "Protein cutoff scanning: A comparative analysis of cutoff dependent and cutoff free methods for prospecting contacts in proteins". *Proteins*, vol. 74, 2009, pp. 727–743.

[SOT00] C. Sotriffer, W. Flader, R. Winger, B. Rode, K. Liedl, J. Varga. "Automated docking of ligands to antibodies: Methods and applications". *Methods*, vol. 20, 2000, pp. 280–291.

[STO86] M. Stonebraker, L.A. Rowe. "The design of PostGres". *SIGMOD Rec.*, vol. 15, 1986, pp. 340–355.

[TAN05] P-N. Tan, M. Steinbach, V. Kumar. "Introduction to data mining". Boston: Addison Wesley, 2005, 2° Edição, 769p.

[TEO03] M.L. Teodoro, L.E. Kavraki. "Conformational flexibility models for the receptor in structure based drug design". *Curr. Pharm. Des.*, vol. 9, 2003, pp. 1419–1431.

[TOR94] A.E. Torda, W.F. van Gunsteren. "Algorithms for clustering molecular dynamics configurations". *J. Comput. Chem.*, vol. 15, 1994, pp. 1331–1340.

[TOT08] M. Totrov, R. Abagyan. "Flexible ligand docking to multiple receptor conformations: A practical alternative". *Current Opin. Struct. Biol.*, vol. 18, 2008, pp. 178–184.

[van90] W.F. van Gunsteren, H.J.C. Berendsen. "Computer simulation of molecular dynamics: Methodology, applications, and perspectives in chemistry". *Angewandte Chemie International Edition in English*, vol. 29, 1990, pp. 992–1023.

[WAL95] A.C. Wallace, R.A. Laskowski, J.M. Thornton. "Ligplot: A program to generate schematic diagrams of protein-ligand interactions". *Protein Eng. Des. Sel.*, vol. 8, 1995, pp. 127–134.

[WAN97] Y. Wang, I.H. Witten. "Inducing model trees for continuous classes". In: 9th European Conf. on Machine Learning Poster Papers, 1997, pp. 128–137.

[WAN04] J.T.L. Wang, M.J. Zaki, H.T.T. Toivonen. "Data Mining in Bioinformatics (Advanced Information and Knowledge Processing)". London: Springer-Verlag, 2004, 1° Edição, 340p.

[WAN07] F. Wang, R. Langley, G. Gulten, L.G. Dover, G.S. Besra, W.R. Jacobs Jr., J.C. Sacchettini. "Mechanism of thioamide drug action against tuberculosis and leprosy". *J. Exp. Med.*, vol. 204, 2007, pp. 73–78.

[WEI04] B. Wei, L. Weaver, A. Ferrari, B. Matthews, B. Shoichet. "Testing a flexible-receptor docking algorithm in a model binding site". *J. Mol. Biol.*, vol. 337, 2004, pp. 1161–1182.

[WIN09] A.T. Winck, K.S. Machado, O. Norberto de Souza, D.D. Ruiz. "FReDD: Supporting mining strategies through a flexible-receptor docking database". *LNBI-LNCS Advances in Bioinformatics and Computational Biology*, vol. 5676, 2009, pp. 143–146.

[WIN10a] A.T. Winck, K.S. Machado, O. Norberto de Souza, D.D. Ruiz. "Supporting intermolecular interaction analyses of flexible-receptor docking simulations". In: IADIS International Conference Applied Computing, 2010, pp. 183–190.

[WIN10b] A.T. Winck, K.S. Machado, D.D. Ruiz, O. Norberto de Souza. "Processo de KDD aplicado à bioinformática" . *Topicos em sistemas colaborativos, multimídia, web e banco de dados. Sociedade Brasileira de Computação*, vol. 1, 2010, pp. 159-180.

[WIN10c] A.T. Winck, K.S. Machado, O. Norberto de Souza, D.D. Ruiz. "A context-based preprocessing on flexible-receptor docking data". In: International Society for Computational Biology Latin America Conference, 2010, pp. 68–68.

[WIN11] A.T. Winck, K.S. Machado, O. Norberto de Souza, D.D. Ruiz. "Context-based preprocessing of molecular docking biological data". *Int. J. Data Min. Bioinform.*, Submetido para revisão, 2011, 20p.

[WIT05] I.H. Witten, E. Frank. "Data Mining: Practical Machine Learning Tools and Techniques". New York: Morgan Kaufmann, 2005 , 2° Edição, 629p.

[WON08] C.F. Wong. "Flexible ligand-flexible protein docking in protein kinase systems". *Biochimica et Biophysica Acta (BBA) - Proteins & Proteomics*, vol. 1784, 2008, pp. 244–251.

[XU08] R. Xu, D.C. Wunsch II. "Clustering". New York: Wiley-IEEE Press, 2008, 1° Edição, 364p.

[YUR10] E. Yuriev, M. Agostino, P.A. Ramsland. "Challenges and advances in computational docking: 2009 in Review". *J. Mol. Recognit.*, vol. 24, 2011, pp. 149–164.

[ZHA07] Y. Zhao, M.F.F. Sanner. "FlipDock: Docking flexible ligands into flexible receptors". *Proteins*, vol. 68, 2007, pp. 726–737.

Apêndice A. Árvores de Decisão Geradas com o Algoritmo J48 e Discretização Método 3

Este apêndice mostra as Figuras que contém as árvores de decisão geradas com o algoritmo J48 para os ligantes PIF, TCL e ETH. As árvores das figuras são as saídas do algoritmo J48 do WEKA.

```
HIE92 <= 9.612197
|   ILE201 <= 5.822563: BOM (44.0/21.0)
|   ILE201 > 5.822563: REGULAR (2973.0/374.0)
HIE92 > 9.612197: MUITO_RUIM (25.0/6.0)
```

Figura A.1: Árvore de decisão para o PIF - Método 3.

```
PHE96 <= 5.375236
|   SER18 <= 6.668699: REGULAR (2261.0/652.0)
|   SER18 > 6.668699
|   |   VAL237 <= 12.063371
|   |   |   GLN65 <= 6.117742: RUIM (55.0/22.0)
|   |   |   GLN65 > 6.117742
|   |   |   |   SER18 <= 7.449048: REGULAR (71.0/29.0)
|   |   |   |   SER18 > 7.449048: RUIM (63.0/41.0)
|   |   VAL237 > 12.063371: REGULAR (52.0/11.0)
PHE96 > 5.375236
|   SER93 <= 2.314529
|   |   ARG194 <= 7.674682: REGULAR (129.0/53.0)
|   |   ARG194 > 7.674682: RUIM (103.0/54.0)
|   SER93 > 2.314529: RUIM (103.0/51.0)
```

Figura A.2: Árvore de decisão para o TCL - Método 3.

```
ILE14 <= 13.369341
|   LEU167 <= 5.054192
|   |   MET154 <= 7.327551
|   |   |   ASP233 <= 9.565454
|   |   |   |   GLY220 <= 12.615396: REGULAR (100.0/26.0)
|   |   |   |   GLY220 > 12.615396
|   |   |   |   |   ILE14 <= 11.080222: REGULAR (67.0/21.0)
|   |   |   |   |   ILE14 > 11.080222: BOM (67.0/24.0)
|   |   |   ASP233 > 9.565454: REGULAR (122.0/21.0)
|   |   MET154 > 7.327551: REGULAR (124.0/20.0)
|   LEU167 > 5.054192: REGULAR (1801.0/325.0)
ILE14 > 13.369341
|   ALA153 <= 3.012108: EXCELENTE (62.0/36.0)
|   ALA153 > 3.012108
|   |   PHE148 <= 2.689391
|   |   |   ARG224 <= 3.166454
|   |   |   |   TYR258 <= 4.225802: EXCELENTE (79.0/42.0)
|   |   |   |   TYR258 > 4.225802
|   |   |   |   |   ILE20 <= 12.138119: BOM (58.0/23.0)
|   |   |   |   |   ILE20 > 12.138119: REGULAR (53.0/30.0)
|   |   |   ARG224 > 3.166454: BOM (159.0/74.0)
|   |   PHE148 > 2.689391
|   |   |   PRO155 <= 8.908839
|   |   |   |   LEU216 <= 9.656394: REGULAR (215.0/83.0)
|   |   |   |   LEU216 > 9.656394: BOM (55.0/23.0)
|   |   |   PRO155 > 8.908839: REGULAR (81.0/50.0)
```

Figura A.3: Árvore de decisão para o ETH - Método 3.

Apêndice B. Árvores Modelo Geradas com o Segundo Conjunto de Experimentos de Regressão com o Algoritmo M5P

Esse apêndice mostra as Figuras B.1, B.2 e B.3 das árvores modelo geradas com o segundo conjunto de experimentos com o algorimto M5P para os ligantes PIF, TCL e ETH.

```
MET146 <= 8.106 :
|   GLY191 <= 3.115 : LM1 (12838/20.278%)
|   GLY191 >  3.115 :
|   |   ASN158 <= 9.491 :
|   |   |   PHE108 <= 16.664 : LM2 (502/30.649%)
|   |   |   PHE108 >  16.664 :
|   |   |   |   PRO111 <= 14.828 :
|   |   |   |   |   GLY118 <= 3.817 : LM3 (237/26.12%)
|   |   |   |   |   GLY118 >  3.817 : LM4 (945/24.416%)
|   |   |   |   PRO111 >  14.828 : LM5 (595/19.677%)
|   |   ASN158 >  9.491 :
|   |   |   ASP147 <= 6.184 :
|   |   |   |   HIE23 <= 7.862 :
|   |   |   |   |   MET129 <= 9.944 :
|   |   |   |   |   |   GLN65 <= 12.056 : LM6 (1533/23.113%)
|   |   |   |   |   |   GLN65 >  12.056 : LM7 (851/21.466%)
|   |   |   |   |   MET129 >  9.944 :
|   |   |   |   |   |   LEU187 <= 7.279 : LM8 (342/22.966%)
|   |   |   |   |   |   LEU187 >  7.279 : LM9 (705/19.408%)
|   |   |   |   HIE23 >  7.862 : LM10 (1227/19.089%)
|   |   |   ASP147 >  6.184 : LM11 (4338/24.877%)
MET146 >  8.106 :
|   ASP63 <= 12.784 :
|   |   SER12 <= 4.633 : LM12 (2071/22.065%)
|   |   SER12 >  4.633 :
|   |   |   ASN66 <= 6.067 : LM13 (284/32.423%)
|   |   |   ASN66 >  6.067 :
|   |   |   |   SER18 <= 5.102 : LM14 (747/25.914%)
|   |   |   |   SER18 >  5.102 : LM15 (374/32.419%)
|   ASP63 >  12.784 :
|   |   MET146 <= 14.054 :
|   |   |   VAL91 <= 18.732 :
|   |   |   |   ASP109 <= 20.672 : LM16 (882/37.738%)
|   |   |   |   ASP109 >  20.672 : LM17 (731/26.965%)
|   |   |   VAL91 >  18.732 : LM18 (429/45.549%)
|   |   MET146 >  14.054 : LM19 (511/44.774%)
```

Figura B.1: Árvore modelo do ligante PIF para o experimento 2.

```
SER12 <= 5.975 :
|   THR38 <= 3.815 :
|   |   GLY118 <= 7.54 : LM1 (3895/19.404%)
|   |   GLY118 >  7.54 :
|   |   |   GLY95 <= 2.209 :
|   |   |   |   LEU206 <= 14.605 : LM2 (379/23.161%)
|   |   |   |   LEU206 >  14.605 : LM3 (825/20.433%)
|   |   |   GLY95 >  2.209 : LM4 (335/20.915%)
|   THR38 >  3.815 :
|   |   VAL170 <= 11.282 : LM5 (5778/21.936%)
|   |   VAL170 >  11.282 :
|   |   |   ALA153 <= 18.633 :
|   |   |   |   SER93 <= 2.309 :
|   |   |   |   |   LEU187 <= 7.967 :
|   |   |   |   |   |   LEU167 <= 5.983 : LM6 (1475/23.142%)
|   |   |   |   |   |   LEU167 >  5.983 : LM7 (462/21.472%)
|   |   |   |   |   LEU187 >  7.967 : LM8 (495/21.997%)
|   |   |   |   SER93 >  2.309 :
|   |   |   |   |   LEU37 <= 8.144 :
|   |   |   |   |   |   LEU206 <= 14.254 : LM9 (511/21.226%)
|   |   |   |   |   |   LEU206 >  14.254 : LM10 (493/21.988%)
|   |   |   |   |   LEU37 >  8.144 :
|   |   |   |   |   |   SER93 <= 2.61 :
|   |   |   |   |   |   |   LEU62 <= 11.255 : LM11 (614/20.515%)
|   |   |   |   |   |   |   LEU62 >  11.255 : LM12 (433/21.792%)
|   |   |   |   |   |   SER93 >  2.61 :
|   |   |   |   |   |   |   SER165 <= 14.026 :
|   |   |   |   |   |   |   |   ALA200 <= 8.756 : LM13 (655/20.894%)
|   |   |   |   |   |   |   |   ALA200 >  8.756 :
|   |   |   |   |   |   |   |   |   ILE256 <= 10.899 : LM14 (605/21.17%)
|   |   |   |   |   |   |   |   |   ILE256 >  10.899 : LM15 (596/20.925%)
|   |   |   |   |   |   |   SER165 >  14.026 : LM16 (822/22.426%)
|   |   |   ALA153 >  18.633 :
|   |   |   |   PHE148 <= 10.934 :
|   |   |   |   |   ARG42 <= 3.848 : LM17 (620/28.342%)
|   |   |   |   |   ARG42 >  3.848 : LM18 (2367/20.513%)
|   |   |   |   PHE148 >  10.934 :
|   |   |   |   |   MET97 <= 5.824 : LM19 (337/25.621%)
|   |   |   |   |   MET97 >  5.824 : LM20 (979/24.358%)
SER12 >  5.975 :
|   SER93 <= 9.076 :
|   |   ILE256 <= 9.723 : LM21 (788/22.185%)
|   |   ILE256 >  9.723 : LM22 (1676/21.431%)
|   SER93 >  9.076 : LM23 (3206/35.583%)
```

Figura B.2: Árvore modelo do ligante TCL para o experimento 2.

```
LEU45 <= 18.648 :
|   ASP147 <= 3.684 :
|   |   SER169 <= 13.21 : LM1 (11661/78.168%)
|   |   SER169 >  13.21 : LM2 (7374/73.631%)
|   ASP147 >  3.684 :
|   |   VAL237 <= 7.653 :
|   |   |   MET231 <= 6.63 : LM3 (938/80.407%)
|   |   |   MET231 >  6.63 :
|   |   |   |   SER12 <= 4.863 :
|   |   |   |   |   ALA21 <= 2.672 :
|   |   |   |   |   |   SER18 <= 5.378 : LM4 (784/69.48%)
|   |   |   |   |   |   SER18 >  5.378 : LM5 (532/67.707%)
|   |   |   |   |   ALA21 >  2.672 : LM6 (332/64.969%)
|   |   |   |   SER12 >  4.863 :
|   |   |   |   |   ASP260 <= 7.476 : LM7 (309/68.84%)
|   |   |   |   |   ASP260 >  7.476 :
|   |   |   |   |   |   GLY13 <= 2.408 : LM8 (454/67.18%)
|   |   |   |   |   |   GLY13 >  2.408 : LM9 (565/63.453%)
|   |   VAL237 >  7.653 :
|   |   |   VAL11 <= 6.754 : LM10 (344/78.114%)
|   |   |   VAL11 >  6.754 :
|   |   |   |   ALA197 <= 4.55 : LM11 (529/58.606%)
|   |   |   |   ALA197 >  4.55 : LM12 (516/62.021%)
LEU45 >  18.648 :
|   ASP149 <= 3.432 :
|   |   PHE148 <= 2.45 : LM13 (911/96.853%)
|   |   PHE148 >  2.45 :
|   |   |   ASP149 <= 2.478 :
|   |   |   |   ALA166 <= 12.807 : LM14 (883/104.834%)
|   |   |   |   ALA166 >  12.807 : LM15 (238/114.406%)
|   |   |   ASP149 >  2.478 : LM16 (1134/95.023%)
|   ASP149 >  3.432 :
|   |   LEU167 <= 4.936 :
|   |   |   TYR157 <= 3.295 : LM17 (858/87.398%)
|   |   |   TYR157 >  3.295 : LM18 (323/81.192%)
|   |   LEU167 >  4.936 : LM19 (1745/85.745%)
```

Figura B.3: Árvore modelo do ligante ETH para o experimento 2.

Apêndice C. Resultados dos Experimentos TCN x RMS - THT

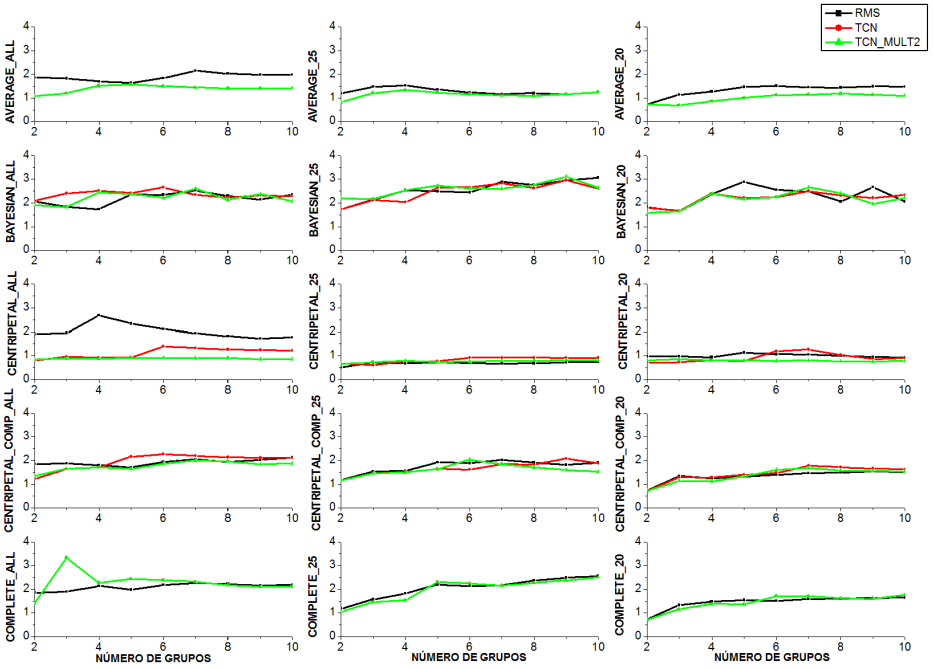


Figura C.1: Resultado da métrica DBI para os algoritmos *Average, Bayesian, Centripetal, Centripetal_Comp* e *Complete* executados com as funções RMS, TCN e TCN_Mult2 com entrada THT.

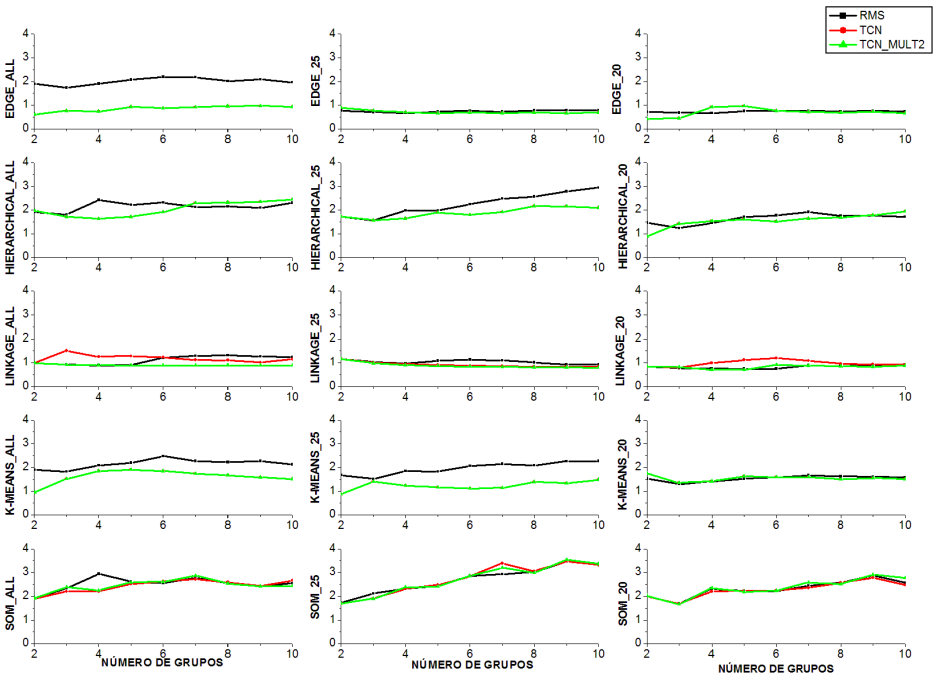


Figura C.2: Resultado da métrica DBI para os algoritmos *Edge, Hierarchical, Linkage, K-means* e *SOM* executados com as funções RMS, TCN e TCN_Mult2 com entrada THT.

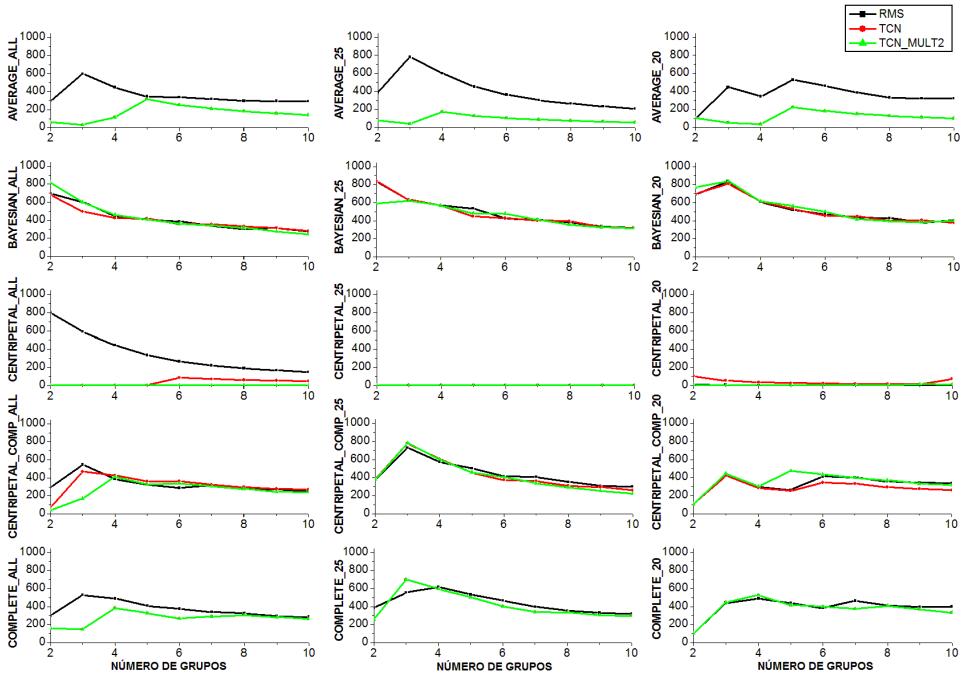


Figura C.3: Resultado da métrica pSF para os algoritmos *Average, Bayesian, Centripetal, Centripetal_Comp* e *Complete* executados com as funções RMS, TCN e TCN_Mult2 com entrada THT.

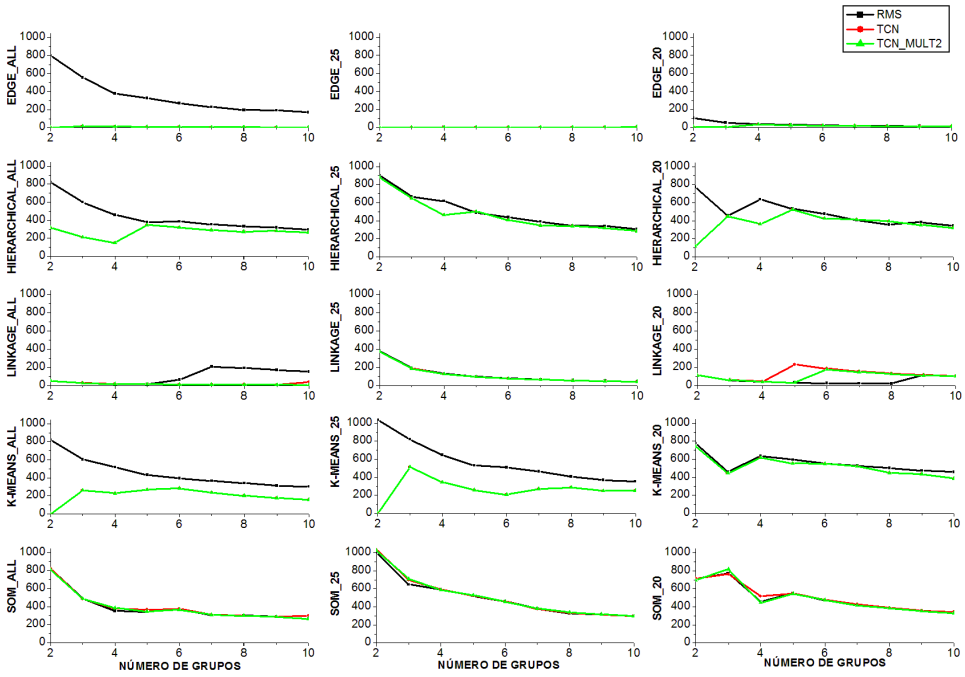


Figura C.4: Resultado da métrica pSF para os algoritmos *Edge, Hierarchical, Linkage, K-means* e *SOM* executados com as funções RMS, TCN e TCN_Mult2 com entrada THT.

Apêndice D. Resultados dos Experimentos CORREL X RMS-THT

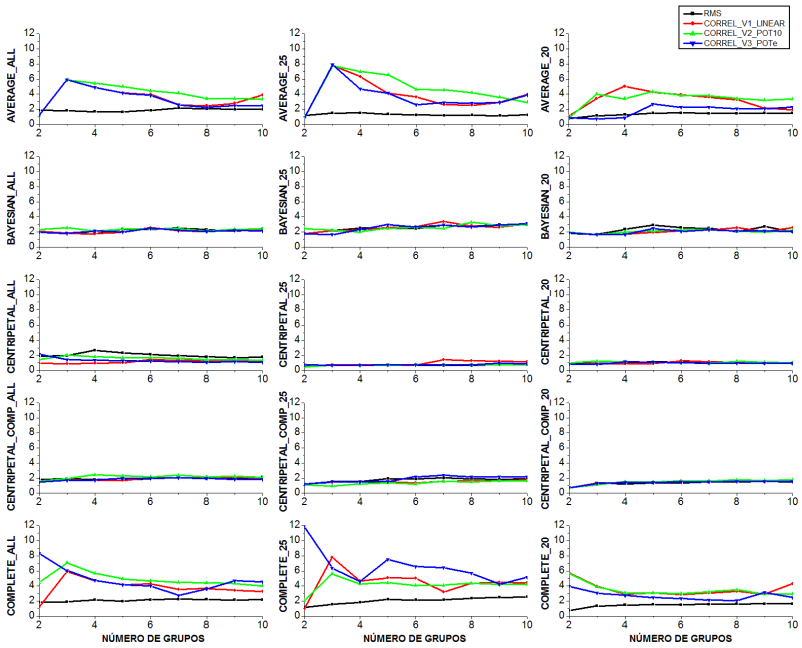


Figura D.1: Resultado da métrica DBI para os algoritmos *Average, Bayesian, Centripetal, Centripetal_Comp* e *Complete* executados com RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ com entrada THT.

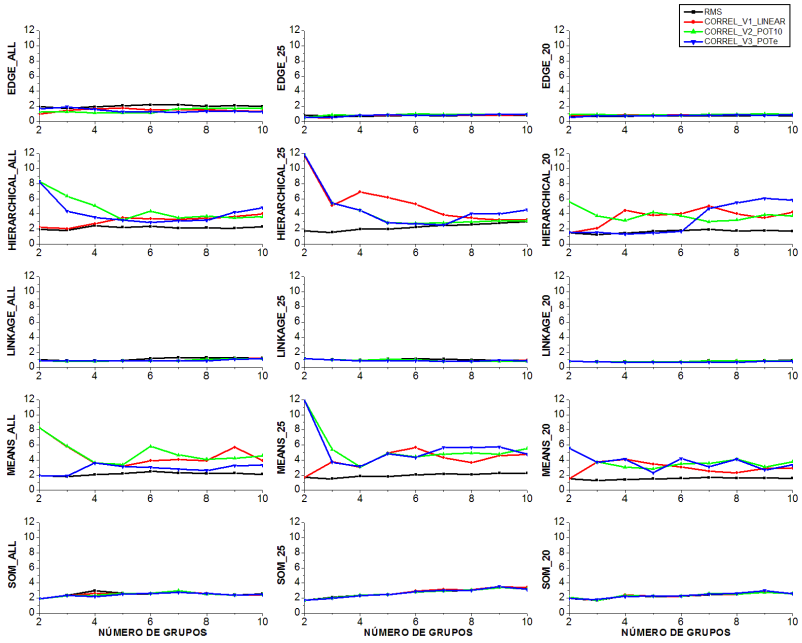


Figura D.2: Resultado da métrica DBI para os algoritmos *Edge, Hierarchical, Linkage, K-means* e *SOM* executados com RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ com THT.

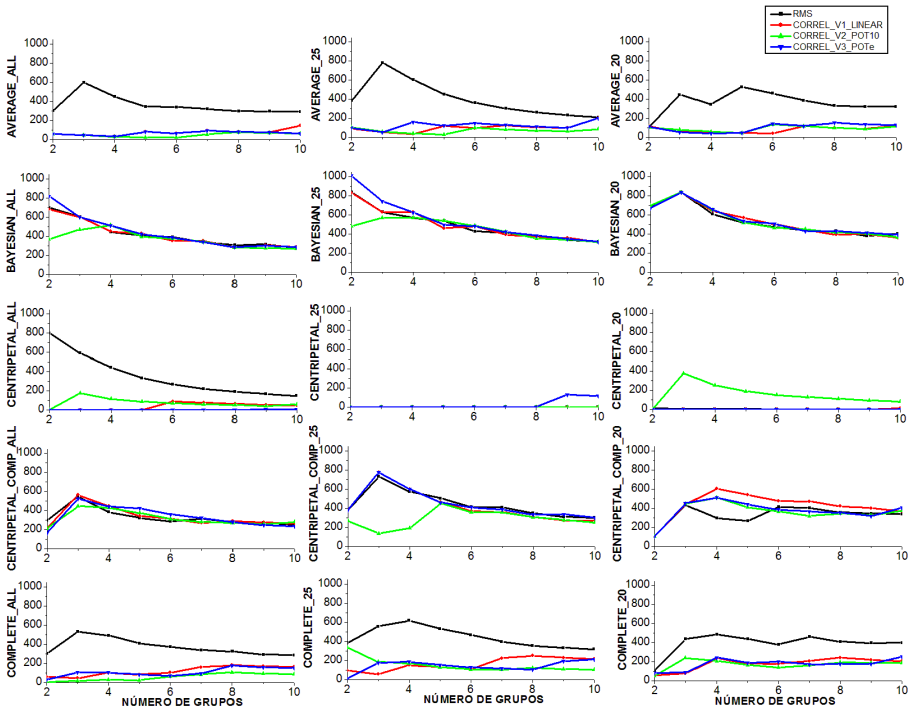


Figura D.3: Resultado da métrica pSF para os algoritmos *Average*, *Bayesian*, *Centripetal*, *Centripetal_Comp* e *Complete* executados com as funções RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ com entrada THT.

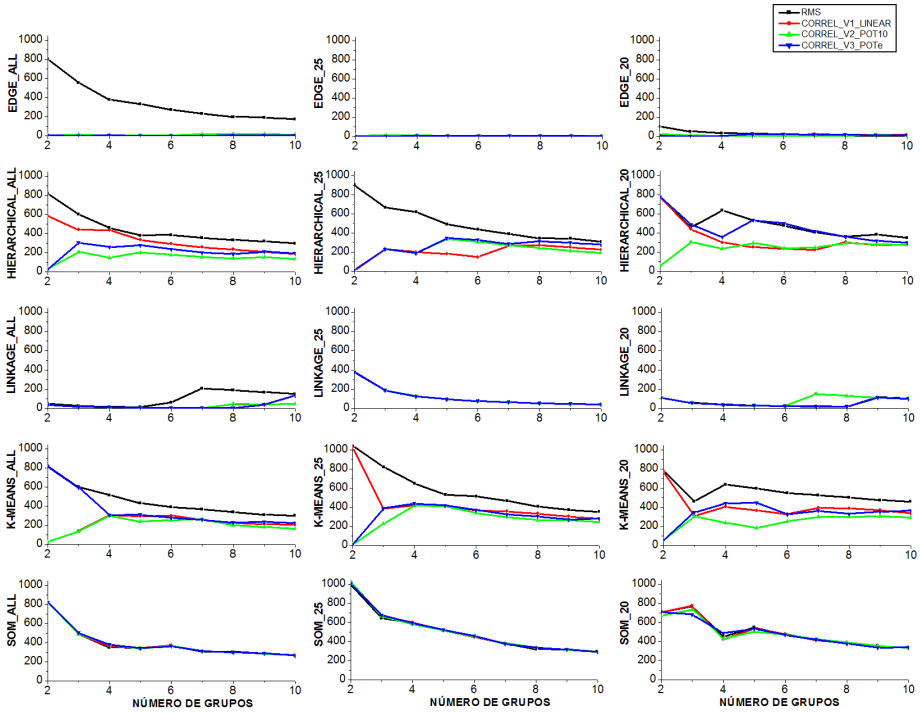


Figura D.4: Resultado da métrica pSF para os algoritmos *Edge*, *Hierarchical*, *Linkage*, *K-means* e *SOM* executados com as funções RMS, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ com entrada THT.

Apêndice E. Resultados dos Experimentos - Avaliações das Médias de Desvio Padrão de FEB Dentro de cada Grupo

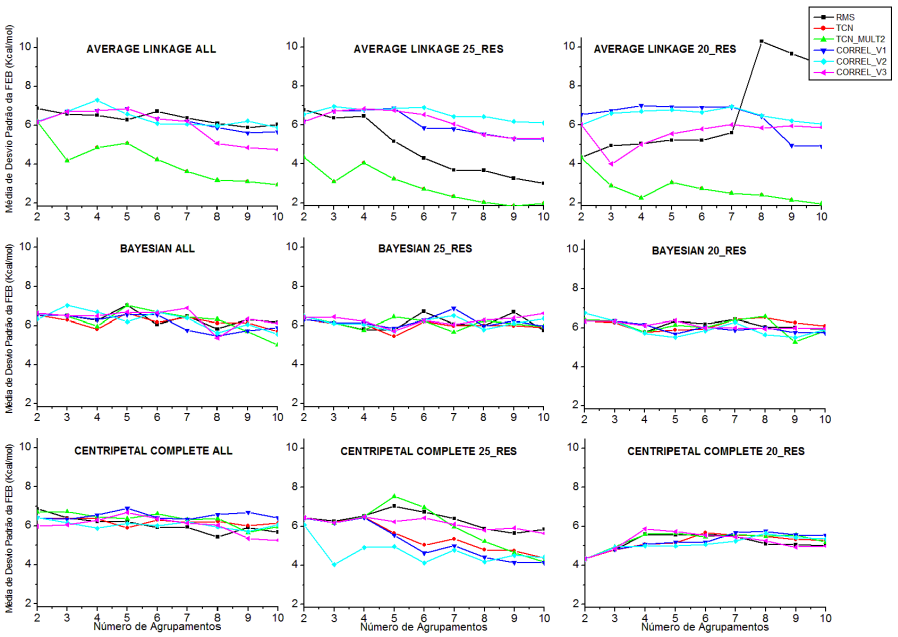


Figura E.1: Média de desvio padrão de FEB para o ligante NADH com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT) para os algoritmos *Average, Bayesian* e *Centripetal_Comp* (ALL, 25_RES e 20_RES).

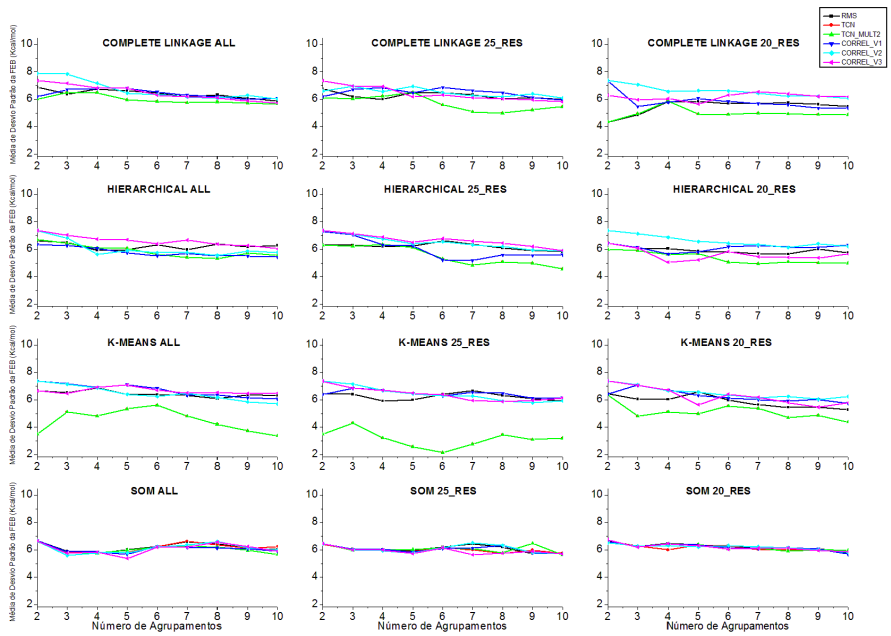


Figura E.2: Média de desvio padrão de FEB para o ligante NADH com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT) para os algoritmos *Complete, Hierarchical, K-means* e *SOM* (ALL, 25_RES e 20_RES).

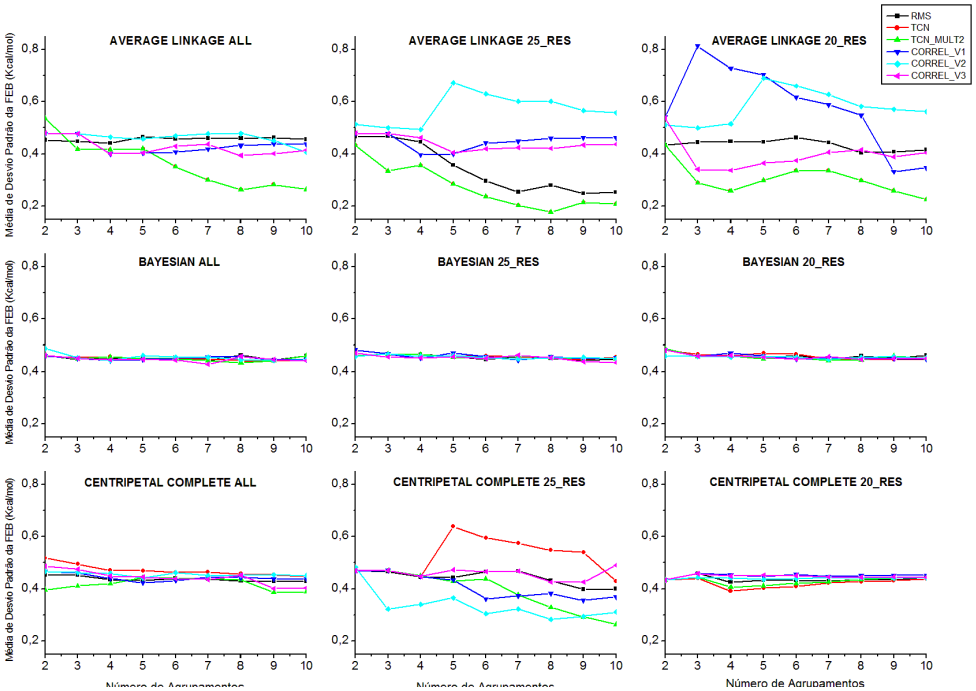


Figura E.3: Média de desvio padrão de FEB para o ligante TCL com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT) para os algoritmos *Average, Bayesian* e *Centripetal_Comp* (ALL, 25_RES e 20_RES).

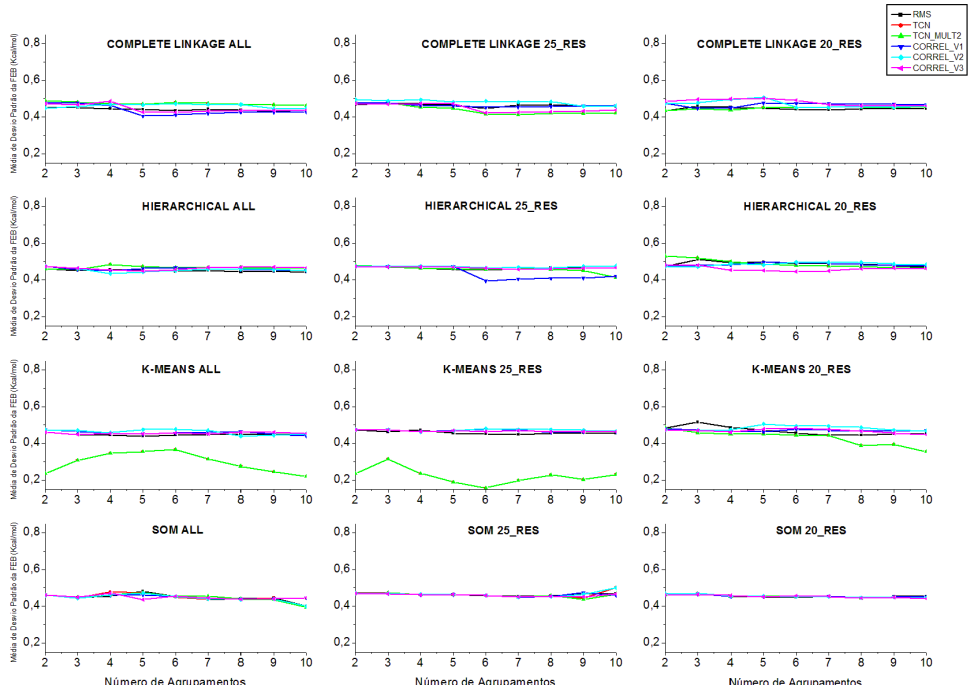


Figura E.4: Média de desvio padrão de FEB para o ligante TCL com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT) para os algoritmos *Complete, Hierarchical, K-means* e *SOM* (ALL, 25_RES e 20_RES).

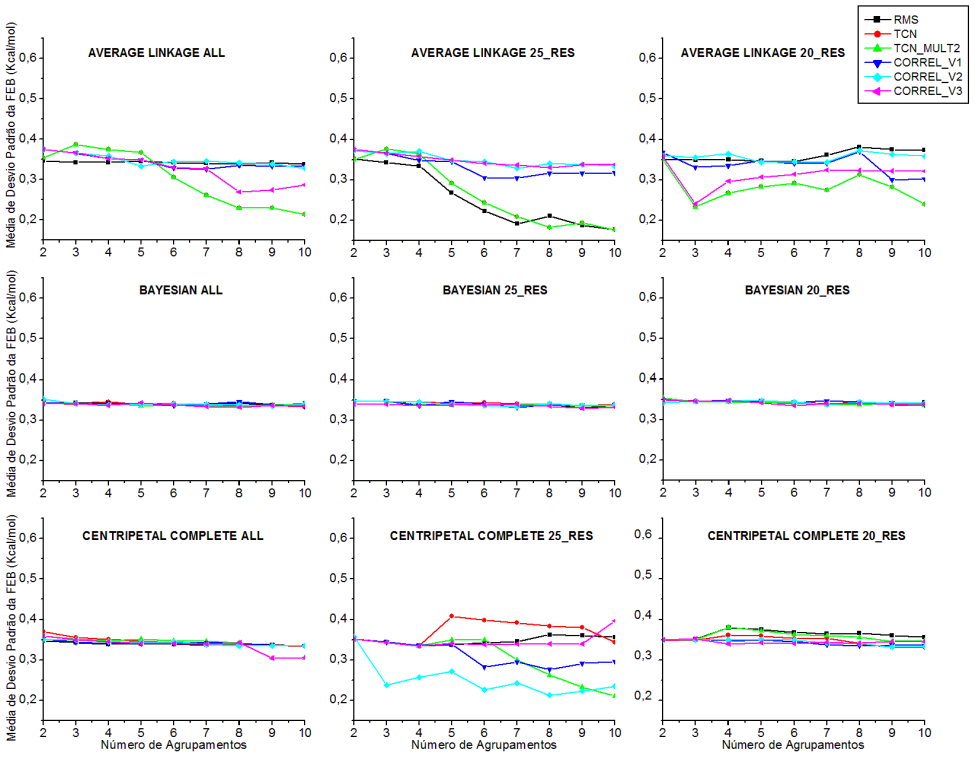


Figura E.5: Média de desvio padrão de FEB para o ligante ETH com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT) para os algoritmos *Average, Bayesian* e *Centripetal_Comp* (ALL, 25_RES e 20_RES).

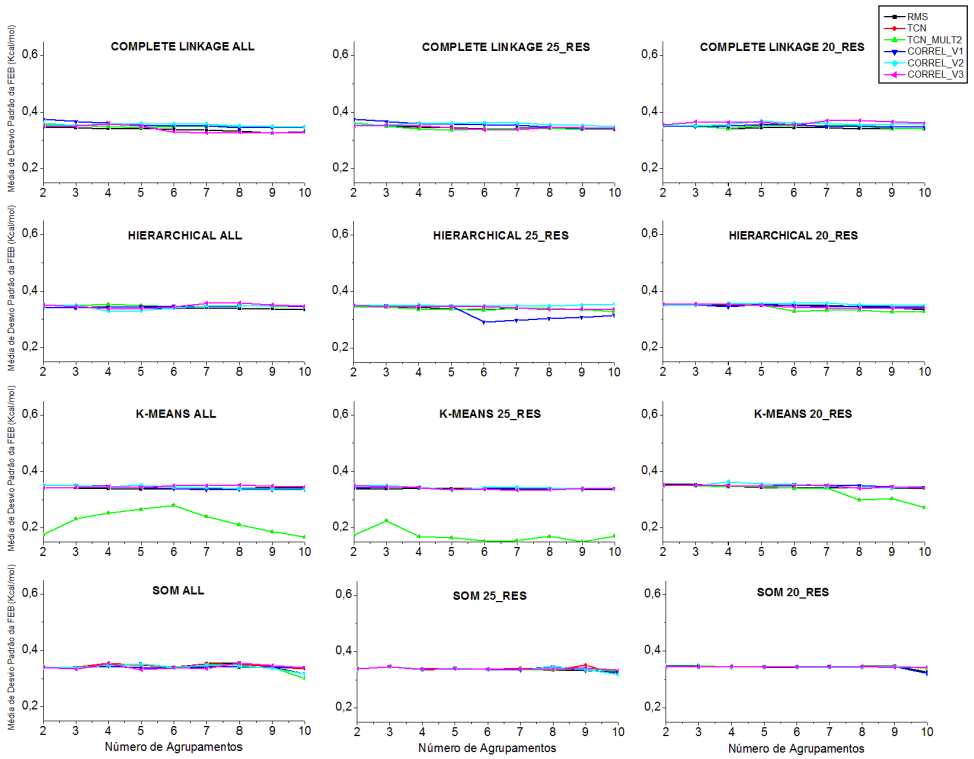


Figura E.6: Média de desvio padrão de FEB para o ligante ETH com as funções de similaridade RMS, TCN, TCN_Mult2, $CORREL\_V1$, $CORREL\_V2$ e $CORREL\_V3$ (entrada THT) para os algoritmos *Complete, Hierarchical, K-means* e *SOM* (ALL, 25_RES e 20_RES).