

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

**3D-Tri: Um Algoritmo de Indução de Árvore de
Regressão para Propriedades Tridimensionais -
um estudo sobre dados de docagem molecular
considerando a flexibilidade do receptor**

ANA TRINDADE WINCK

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. Duncan Dubugras Alcoba Ruiz

**Porto Alegre
2012**

W761t Winck, Ana Trindade
3D-Tri : um algoritmo de indução de árvore de regressão
para propriedades tridimensionais : um estudo sobre dados de
docagem molecular considerando a flexibilidade do receptor /
Ana Trindade Winck. – Porto Alegre, 2012.
108 f.

Tese (Doutorado) – Fac. de Informática, PUCRS.
Orientador: Prof. Dr. Duncan Dubugras Alcoba Ruiz.

1. Bioinformática. 2. Mineração de Dados.
3. Dinâmica Molecular. I. Ruiz, Duncan Dubugras Alcoba.
II. Título.


CDD 005.74

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



TERMO DE APRESENTAÇÃO DE TESE DE DOUTORADO

Tese intitulada "3D-Tri: Um Algoritmo de Indução de Árvore de Regressão para Propriedades Tridimensionais - um estudo sobre dados de docagem molecular considerando a flexibilidade do receptor", apresentada por Ana Trindade Winck, como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação, Sistemas de Informação, aprovada em 17/01/2012 pela Comissão Examinadora:


Prof. Dr. Duncan Dubugras Alcoba Ruiz -
Orientador

PPGCC/PUCRS


Profa. Dra. Karin Becker -

UFRGS

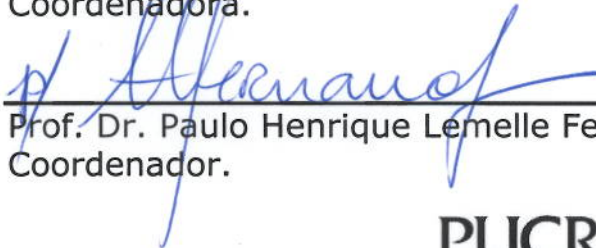

Prof. Dr. Adriano Velasque Werhli -

FURG


Prof. Dr. Osmar Norberto de Souza -

PPGCC/PUCRS

Homologada em 27/01/2012, conforme Ata No. 003/2012 pela Comissão Coordenadora.


Prof. Dr. Paulo Henrique Lemelle Fernandes
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

À minha família.

AGRADECIMENTOS

A trajetória percorrida em um doutorado não é fácil. Chegar ao final com êxito só é possível com a companhia, compreensão e incentivo de todos. Agradecer é a maneira de expressar a minha gratidão e reconhecer a importância de cada um para o desenvolvimento desta Tese.

O primeiro - e mais especial - muito obrigada devo à minha família, principalmente aos meus pais Irani e Rosecler. Sempre presentes, amorosos e incentivadores em todas as minhas decisões, são os principais responsáveis por eu ser quem eu sou hoje. Tenho um grande orgulho de ser sua filha. Amo muito vocês! Agradeço também à minha vó Noely e à minha dinda Rosângela, sempre felizes e orgulhosas de minhas vitórias. Ao meu dindo Geraldo, também feliz com minhas conquistas. Um muito obrigada aos meus tios Didi, Édison e Venilda, e aos meus primos queridos que acompanharam de perto essa trajetória. Se a família é a base de tudo, eu estou muito bem apoiada!

Um agradecimento muito especial ao meu orientador Duncan D. Ruiz, por ter acreditado no meu trabalho e compartilhado seus conhecimentos durante toda pós-graduação. Obrigada pelos ensinamentos, companheirismo e pela oportunidade de ter sido tua orientada. Agradeço ao professor Osmar pela troca de conhecimento e pela oportunidade de pesquisa em bioinformática. À professora Vera por ter me proporcionado o primeiro estágio sanduíche na USP São Carlos e ao professor André por ter-me recebido nessa instituição. Novamente ao professor Duncan pela oportunidade do segundo estágio sanduíche e ao professor Aad van Moorsel por me acolher na Newcastle University. Todas essas experiências foram fundamentais para a minha formação como pessoa e pesquisadora.

Aos colegas e amigos do GPIN, em especial Christian Quevedo, Juliano Varella, Leandro Bogoni, Luciano Blomberg, Nelson Tenório, Patrícia Hubler e Rodrigo Barros pelos ótimos momentos de convivência, troca de conhecimento e compartilhamento das mais variadas idéias. Aos amigos e colegas do LaBIO, principalmente ao Anderson, André, Carla, Danieli e Elisa pelos momentos que passamos juntos. Agradeço também aos amigos do LABIC e BIOCOM, por me receber bem e por ainda cultivarem uma amizade e carinho especial. Foram momentos inesquecíveis que passei junto a vocês. Agradeço também aos colegas e amigos de Newcastle. Sinto falta de todos.

Um parágrafo especial dedico a três grandes amigos, Karina Machado, Márcio Basgalupp e Tiago Silva, com os quais eu dividi os momentos mais importantes da vida acadêmica e pessoal durante a pós-graduação. A Karina é uma incansável pesquisadora e uma amiga incondicional. Com ela cultivei uma grande amizade, aprendi a trabalhar em conjunto e aprendi que parcerias são fundamentais para crescermos como pessoas e profissionais. Sem ela esta Tese não teria atingido todos os resultados que obtivemos. O Márcio é um amigo especial em quem me espelho ao agir com determinação para conseguir aquilo que realmente quero. O Tiago é um grande amigo que esteve junto em todos os momentos, compartilhando conquistas, dúvidas, alegrias e lágrimas, sempre com incentivo para que tudo desse certo. Muito obrigada a vocês, queridos amigos.

Por fim, agradeço à todos do PGGCC, colegas, professores e funcionários, pela convivência durante esses anos. Ao CNPq agradeço pela concessão da bolsa de doutorado, fundamental para o desenvolvimento desta Tese. Aos que eu não mencionei recebam o meu sincero muito obrigada.

3D-Tri: Um Algoritmo de Indução de Árvore de Regressão para Propriedades Tridimensionais - um estudo sobre dados de docagem molecular considerando a flexibilidade do receptor

RESUMO

Com o avanço nos experimentos biológicos, a manipulação e análise do grande volume de dados sendo gerados por esses experimentos têm sido um dos desafios em bioinformática, onde uma importante área de pesquisa é o desenho racional de fármacos (RDD - *Rational Drug Design*). A interação entre macromoléculas biológicas, chamadas de receptores, e pequenas moléculas, chamadas ligantes, é o princípio fundamental do RDD. É em experimentos *in silico* de docagem molecular que se investiga o melhor encaixe e conformação de um ligante em uma cavidade do receptor. O resultado de um experimento de docagem pode ser avaliado a partir de um valor contínuo de energia livre de ligação (FEB - *Free Energy of Binding*). Tem-se empregado esforços em minerar dados de resultados de docagem molecular, com o objetivo de selecionar conformações relevantes para reduzir o tempo de futuros experimentos de docagem. Nesse sentido, foi desenvolvido um repositório para armazenar todos os dados a respeito desses experimentos, em nível de detalhe. Com esse repositório, os dados foram devidamente pré-processados e submetidos a diferentes tarefas de mineração de dados. Dentre as técnicas aplicadas, a que apresentou-se mais promissora para o tipo de dados sendo utilizado foi árvore de decisão para regressão. Apesar dos resultados alcançados por esses experimentos serem promissores, existem algumas propriedades nos experimentos que dificultam a efetiva seleção de conformações. Dessa forma, propõe-se uma estratégia que considera as propriedades tridimensionais (3D) do receptor para prever o valor de FEB. Assim, nesta Tese é apresentado o 3D-Tri, um algoritmo de indução de árvore de regressão que considera essas propriedades 3D, onde essas propriedades são definidas como atributos no formato x, y, z . O algoritmo proposto faz uso dessas coordenadas para dividir um nodo em duas partes, onde o átomo sendo testado para o nodo é avaliado em termos de sua posição em um bloco $[(x_i, x_f); (y_i, y_f); (z_i, z_f)]$ que melhor represente sua posição no espaço, onde i indica a posição inicial de uma coordenada, e f indica a posição final. O modelo induzido pode ser útil para um especialista de domínio para selecionar conformações promissoras do receptor, tendo como base as regiões dos átomos que aparecem no modelo e que indicam melhores valores de FEB.

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

3D-Tri: A Regression Decision Tree Induction Algorithm for Threedimensional Properties - a study on flexible-receptor molecular docking data

ABSTRACT

With the growth of biological experiments, solving and analyzing the massive amount of data being generated has been one of the challenges in bioinformatics, where one important research area is the rational drug design (RDD). The interaction between biological macromolecules called receptors, and small molecules called ligands, is the fundamental principle of RDD. In in-silico molecular docking experiments it is investigated the best bind and conformation of a ligand into a receptor. A docking result can be discriminated by a continue value called Free Energy of Binding (FEB). We are attempting on mining data from molecular docking results, aiming at selecting promising receptor conformations to the next docking experiments. In this sense, we have developed a comprehensive repository to store our molecular docking data. Having such repository, we were able to apply preprocessing strategies on the stored data and submit them to different data mining tasks. Among the techniques applied, the most promising results were obtained with regression model trees. Although we have already addressed important issues and achieved significant results, there are some properties in these experiments turning it difficult to properly select conformations. Hence, a strategy was proposed considering the three-dimensional (3D) properties of the receptor conformations, to predict FEB. This thesis presents the 3D-Tri, a novel algorithm able to handle and treat spatial coordinates in a x, y, z format, and induce a tree that predicts FEB value by representing such properties. The algorithm uses such coordinates to split a node in two parts, where the edges evaluate whether the atom being tested by the node is part of a given interval $[(x_i, x_f); (y_i, y_f); (z_i, z_f)]$, where i indicates the initial position of the coordinate, and f its final position. The induced model can help a domain specialist to select promising conformations, based on the region of the atoms in the model, to perform new molecular docking experiments.

Keywords: Data Mining, 3D properties, Molecular Docking, Flexible Receptor, Molecular Dynamics.

LISTA DE FIGURAS

Figura 2.1	Conjunto de dados e sua respectiva árvore de decisão. Adaptado de [ALP10]	37
Figura 3.1	Grade 3D considerando o receptor InhA e o ligante PIF.	45
Figura 3.2	Parte do arquivo PDB, que corresponde a 1.0 ps da trajetória de simulação por DM da enzima InhA.	46
Figura 3.3	Parte da conformação 3D do modelo FFR da enzima InhA. Cada cor representa uma conformação distinta. A estrutura cristalográfica (PDB ID: 1ENY) obtida do Protein Data Bank (PDB) [BER00] está representada em laranja; as outras quatro estruturas são conformações médias que variam de 0.0 a 500 ps (em ciano), de 500 a 1.000 ps (em azul), de 1.050 a 1.500 ps (em magenta) e de 1.550 a 2.000 ps (em verde).	47
Figura 3.4	Representações das estruturas 3D dos ligantes NADH (a), PIF (b), TCL (c) e ETH (d).	48
Figura 3.5	Parte do arquivo MOL2 para o ligante NADH.	48
Figura 3.6	Parte do resultado do AutoDock3.0.5, cujo experimento considerou uma conformação da enzima InhA e o ligante TCL.	49
Figura 4.1	Modelo de dados do repositório FReDD.	52
Figura 4.2	Algumas distâncias atômicas entre o ligante PIF e o resíduo GLY95 do receptor InhA.	55
Figura 4.3	Top 10 resíduos do receptor InhA que mais interagem com cada um dos ligantes NADH, PIF, TCL e ETH. O receptor é a estrutura cinza na forma de <i>Ribbons</i> . Os 10 resíduos que mais interagem com cada ligante estão na forma de esfera de <i>van der Waals</i> e os ligantes na forma de palitos.	58
Figura 5.1	Árvore de decisão gerada a partir do arquivo discretizado pelo Método 3 para o complexo InhA-NADH.	66
Figura 5.2	Árvore de decisão gerada a partir do arquivo pré-processado pela Estratégia 3, para o complexo InhA-NADH.	69
Figura 6.1	Divisão de um nodo pelo algoritmo 3D-Tri	79
Figura 6.2	Árvore binária induzida	84
Figura 7.1	Árvore induzida para os top 10 resíduos do ligante ETH pelo algoritmo 3D-Tri	92
Figura 7.2	Árvore induzida para os top 10 resíduos do ligante ETH pelo algoritmo M5P	93

LISTA DE TABELAS

Tabela 3.1	Nomes, abreviaturas e número de átomos dos ligantes sendo utilizados	47
Tabela 4.1	Descrição dos dados contidos nas tabelas do repositório FReDD	53
Tabela 4.2	População do repositório FReDD	53
Tabela 4.3	Distribuição do valor de FEB para cada ligante	54
Tabela 4.4	Parte da matriz [Result] gerada para o ligante PIF	57
Tabela 4.5	Parte da matriz [Result] gerada para o ligante PIF	58
Tabela 4.6	Top 10 resíduos (destacados) para todos os ligantes e suas frequências, totalizando em 25 resíduos. Os <i>top10</i> resíduos para cada ligante estão destacados.	59
Tabela 4.7	Comparação do número de resíduos que interagem com cada ligante na estrutura cristalográfica e com os top10 do modelo FFR.	59
Tabela 5.1	Exemplos de regras de associação extraídas dos experimentos	62
Tabela 5.2	Distribuição de exemplos nas classes para cada método e cada ligante. . . .	64
Tabela 5.3	Resultados dos modelos de árvore de decisão para classificação.	65
Tabela 5.4	Número de atributos selecionados a partir do algoritmo CFS (Estratégia 2) .	68
Tabela 5.5	Número de atributos selecionados a partir do Algoritmo 5.1 (Estratégia 3) .	68
Tabela 5.6	Número de atributos selecionados a partir da combinação das estratégias de seleção de atributos (Estratégia 4)	69
Tabela 5.7	Avaliação do modelo - métricas preditivas	71
Tabela 5.8	Avaliação do modelo - métricas de contexto	71
Tabela 5.9	Análise dos modelos lineares	73
Tabela 6.1	Exemplo de um conjunto de dados gerado para o ligante PIF	78
Tabela 6.2	Dataset fictício para uma propriedade tridimensional	81
Tabela 6.3	Erro calculado para as instâncias ordenadas pela distância Euclidiana	82
Tabela 6.4	Critério de parada para atualização do bloco. Elimina-se a linha 6 e o bloco é atualizado com os valores mínimos e máximos das linhas 1 a 5 para cada coordenada.	83
Tabela 7.1	Exemplo de coordenadas para <i>DatasetETH</i>	89
Tabela 7.2	Métricas preditivas para os modelos induzidos.	93
Tabela 7.3	Métricas de contexto para os modelos induzidos.	94

LISTA DE ALGORITMOS

Algoritmo 2.1	Pseudo-código do K-means. Adaptado de [TAN05] e [HAR79]	35
Algoritmo 2.2	Arvore de decisão para classificação, adaptado de [ALP10].	39
Algoritmo 2.3	Arvore de decisão para regressão, adaptado de [ALP10] e [WAN97].	41
Algoritmo 4.1	Geração do conjunto de dados inicial.	56
Algoritmo 5.1	Seleção de atributos baseada no contexto de dados de docagem molecular.	68
Algoritmo 6.1	Geração de um conjunto de dados 3D.	78
Algoritmo 6.2	Definição do Bloco.	85
Algoritmo 6.3	Indução da Árvore.	87

LISTA DE ABREVIATURAS E SIGLAS

3D	Tridimensional
CFS	<i>Correlation-based Feature Selection</i>
DM	Dinâmica Molecular
DP	Desvio Padrão
DW	<i>Data Warehouse</i>
ER	Entidade-Relacionamento
ETH	Etionamida
FEB	<i>Free Energy of Binding</i> ou Energia Livre de Liga?ão
FFR	<i>Fully Flexible Receptor</i> ou Receptor completamente flexível
FN	Falso Negativo
FP	Falso Positivo
FReDD	<i>Flexible Receptor Docking Database</i>
InhA	Enzima <i>2-trans-enoil ACP(CoA) Redutase</i> de <i>Mycobacterium tuberculosis</i>
KDD	<i>Knowledge Discovery in Databases</i>
LM	<i>Linear Model</i> ou Modelo Linear
MAE	<i>Mean Absolute Error</i> ou Erro Médio Absoluto
MTB	<i>Mycobacterium tuberculosis</i>
OLAP	<i>On-line Analytical Process</i>
PDB	<i>Protein Data Bank</i>
PIF	Isoniazida pentacianoferrato
NADH	Nicotinamida Adenina Dinucleotídeo, forma reduzida
RDD	<i>Rational Drug Design</i> ou Desenho Racional de Fármacos
RDP	Redução do Desvio Padrão
RMSD	<i>Root Mean Square Deviation</i>
RMSD	<i>Root Mean Square Error</i>
SVM	<i>Support Vector Machine</i>
TCL	Triclosano
TEB	Taxa de Excelente e Bons
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1. INTRODUÇÃO	27
1.1 Motivação	28
1.2 Objetivos	28
1.2.1 Objetivo geral	28
1.2.2 Objetivos específicos	29
1.3 Organização da tese	29
2. PROCESSO DE KDD E MINERAÇÃO DE DADOS	31
2.1 Repositório alvo	31
2.1.1 Pré-Processamento	32
2.2 Mineração de dados	33
2.2.1 Regras de Associação	33
2.2.2 Agrupamento	34
2.2.3 Predição	35
2.2.3.1 Indução de árvores de decisão	35
2.2.3.2 Indução de árvores de decisão para classificação	37
2.2.3.3 Indução de árvores de decisão para regressão	40
2.3 Considerações do capítulo	42
3. DOCAGEM MOLECULAR CONSIDERANDO O RECEPTOR FLEXÍVEL	43
3.1 Desenho racional de fármacos	43
3.1.1 Docagem molecular e simulações por dinâmica molecular	44
3.2 Aquisição de dados	46
3.2.1 Receptor	46
3.2.2 Ligantes	47
3.2.3 Experimentos de docagem molecular	48
3.3 Considerações do capítulo	49
4. REPOSITÓRIO ALVO PARA PRÉ-PROCESSAMENTO	51
4.1 O repositório FReDD	52
4.1.1 População do repositório	53
4.2 FReDD como uma infraestrutura para pré-processamento	54
4.2.1 Escolha do atributo alvo	54
4.2.2 Escolha dos atributos preditivos	55

4.2.3	Geração do arquivo de entrada	56
4.3	Análises sobre os dados armazenados no repositório	57
4.4	Considerações do capítulo	60
5.	EXPERIMENTOS COM MINERAÇÃO DE DADOS	61
5.1	Experimentos com regras de associação	62
5.2	Experimentos com árvores de decisão para classificação	63
5.2.1	Discretização do atributo alvo - FEB	63
5.2.2	Avaliação dos modelos induzidos	65
5.3	Experimentos com árvores de decisão para regressão	67
5.3.1	Estratégias de pré-processamento	67
5.3.2	Avaliação dos modelos induzidos	70
5.3.3	Pós-processamento dos modelos induzidos	72
5.4	Considerações sobre os modelos induzidos	73
5.5	Considerações do capítulo	74
6.	ALGORITMO 3D-Tri	77
6.1	Pré-processamento dos dados	77
6.2	Algoritmo	79
6.2.1	Definição do bloco	80
6.2.2	Indução da árvore	84
6.3	Considerações do capítulo	86
7.	TESTE DO ALGORITMO 3D-Tri	89
7.1	Dados utilizados	89
7.2	Plano de teste	90
7.2.1	Indução do modelo a partir do algoritmo 3D-Tri	90
7.2.2	Indução do modelo a partir do algoritmo M5P	91
7.2.3	Avaliação dos modelos	91
7.3	Resultados	92
7.4	Considerações do capítulo	95
8.	TRABALHOS RELACIONADOS	97
8.1	Banco de dados integrado para RDD	97
8.2	Banco de dados para informações tridimensionais de moléculas	97
8.3	Deteccção de contatos atômicos em estruturas tridimensionais	98
8.4	Considerações do Capítulo	98

9. CONCLUSÃO	99
9.1 Publicações	100
9.2 Trabalhos futuros	101
REFERÊNCIAS	103

1. INTRODUÇÃO

Mineração de dados, de acordo com Tan et al. [TAN05] é um processo de descoberta de padrões úteis em grandes repositórios de dados. Essa é uma das etapas do processo de descoberta de conhecimento em bases de dados (KDD - *Knowledge Discovery in Databases*) proposto por Fayyad et al. [FAY96]. Algoritmos de aprendizado de máquina são implementados para a identificação desses padrões. Em problemas preditivos de aprendizado de máquina existem basicamente um conjunto de dados de entrada e uma saída onde a tarefa é aprender como mapear o conjunto de entrada para a saída.

Ainda que existam diversos algoritmos desenvolvidos para atender problemas de predição, muitos desses apenas constroem uma função preditiva que indica a qual valor alvo os objetos minerados pertencem. Entretanto, em alguns problemas de mineração de dados se faz necessário entender o modelo induzido. Conforme Freitas et al. [FRE10], apesar da falta de consenso na literatura em mineração de dados a respeito das tarefas que produzem resultados mais compreensíveis, existe um acordo que representações na forma de árvore de decisão e conjuntos de regras podem ser melhor compreendidos por usuários finais do que representações do tipo caixa-preta, como SVM (*Support Vector Machine*) ou Redes Neurais. Árvores de decisão têm a vantagem de representar o conhecimento descoberto na forma de um grafo, sendo que sua estrutura hierárquica é capaz de apontar a importância dos atributos utilizados para predição.

Existem várias áreas de aplicação onde a construção de um modelo compreensível se faz necessário. Em bioinformática, apenas um conjunto de dados e um conjunto de resultados provenientes da execução de algoritmos de mineração de dados podem não ser suficientes. É preciso que tanto os dados quanto os resultados obtidos representem de forma satisfatória o contexto ao qual eles fazem parte, de modo com que um especialista de domínio possa utilizar e criticar o modelo. Este trabalho está inserido no contexto de Desenho Racional de Fármacos (RDD - *Rational Drug Design*) [KUN92], onde o princípio fundamental diz respeito à interação entre macromoléculas - chamadas receptores - e pequenas moléculas - chamadas de ligantes [LYB95]. É nos experimentos *in-silico* de docagem molecular que se investiga e avalia a melhor ligação de um determinado ligante nas diferentes conformações que um dado receptor pode ter. Tal ligação é avaliada através de uma medida chamada energia livre de ligação (FEB - *Free Energy of Binding*). Os ligantes que obtiverem os melhores resultados são testados em experimentos *in-vitro*. Se o resultado for promissor, um novo medicamento pode ser gerado.

O processo de docar o ligante na estrutura alvo não é uma tarefa trivial. Um dos fatores que influencia nos resultados é a flexibilidade do receptor. Apesar disso, a maioria dos algoritmos que executam docagem molecular somente considera a flexibilidade do ligante, considerando o receptor como uma estrutura rígida. Dentre vários trabalhos que incorporam a flexibilidade do receptor, nesta Tese utiliza-se uma série de experimentos de docagem molecular, considerando em cada experimento uma conformação do receptor gerada por uma simulação de dinâmica molecular (DM) [LIN02].

1.1 Motivação

O principal problema na utilização de uma trajetória por DM refere-se ao tempo necessário para a execução de todos os experimentos, bem como à grande quantidade de dados gerados. O nosso interesse no desenvolvimento deste trabalho está em minerar dados de resultados de experimentos de docagem molecular, a fim de obter modelos que auxiliem na seleção de conformações promissoras do receptor para futuros experimentos de docagem molecular e, assim, contribuir para a redução no tempo desses experimentos. Para tanto, foram empregados esforços em construir um processo completo de KDD para tratar o grande volume de dados envolvido em seus diferentes tipos [WIN10b]. Desta forma, foi desenvolvido um repositório de dados suficientemente abrangente para armazenar os diferentes dados envolvidos em experimentos de docagem molecular [WIN09] [WIN10a]. De posse desse repositório, foi possível aplicar estratégias de pré-processamento sobre os dados armazenados [WIN10c] [WIN11] [MAC10c] e submetê-los a diferentes tarefas de mineração [MAC11], como regras de associação [MAC08], árvores de decisão para classificação [MAC10b] e árvores de decisão para regressão [WIN10c] [WIN11] [MAC10a] [MAC10d].

Dentre as técnicas aplicadas, os resultados que se mostraram mais promissores foram obtidos a partir de árvores de decisão para regressão. No entanto, apesar desses trabalhos mostrarem resultados significativos, acredita-se que eles servem como base e motivação para evoluir a abordagem sendo aplicada e, assim, produzir modelos melhores e mais bem acurados. A abordagem desenvolvida inicialmente utiliza como dados de entrada as diferentes conformações do receptor e a distância dos átomos do resíduo desse receptor em relação aos átomos de um dado ligante. Como modelo de saída, são induzidas árvores que mapeiam resíduos do receptor, indicando qual o melhor intervalo de distância em relação ao ligante que podem produzir melhores valores de FEB.

Em direção à uma nova estratégia de mineração neste contexto, busca-se considerar como dados de entrada as propriedades tridimensionais de cada átomo do receptor para prever um dado valor de FEB, para um dado ligante. Nesse sentido, este trabalho apresenta um novo algoritmo de indução de árvore de regressão capaz de identificar as propriedades tridimensionais inerentes ao problema e induzir uma árvore que indique as melhores posições no espaço Euclidiano de determinados átomos que possam resultar em importantes resultados de FEB e, assim, contribuir para a efetiva seleção de conformações do receptor.

1.2 Objetivos

1.2.1 Objetivo geral

O objetivo desta Tese é definir o algoritmo 3D-Tri (*Three-Dimensional Regression Tree Induction Algorithm*), um novo algoritmo de indução de árvore de regressão para propriedades tridimensionais, onde o algoritmo seja capaz de ler um conjunto de dados provenientes de resultados de simulação por DM, tendo como instâncias as diferentes conformações de um receptor, como atributos preditivos as coordenadas espaciais dos átomos deste receptor, e como atributo alvo o valor de FEB. Além de

apresentar uma maneira de interpretar os atributos do arquivo de entrada como parte de coordenadas em um espaço euclidiano, este algoritmo também se destaca na forma de indução desses atributos, introduzindo uma abordagem de definição de um intervalo ideal para cada coordenada dos átomos envolvidos.

1.2.2 Objetivos específicos

- Criar uma rotina de pré-processamento dos dados de simulação por dinâmica molecular e experimentos de docagem molecular para gerar um conjunto de dados contendo as coordenadas espaciais dos átomos para cada conformação do receptor e os respectivos valores de FEB para um dado ligante;
- Desenvolver uma estratégia de identificação de um intervalo com pontos iniciais e finais das coordenadas de cada átomo, cuja estrutura pode ser graficamente representada na forma de um bloco;
- Estabelecer uma abordagem de indução de árvores binárias para o intervalo, ou bloco, identificado para cada átomo;
- Definir um algoritmo de indução de árvore de regressão que utilize as propriedades citadas nos objetivos descritos anteriormente, e mostrar a qualidade dos mesmos.

1.3 Organização da tese

Esta Tese está organizada conforme segue:

- O Capítulo 2 apresenta conceitos de mineração de dados, com foco nos algoritmos tradicionais de indução de árvores de decisão, tanto para classificação quanto para regressão. São apresentadas as principais estratégias empregadas por cada algoritmo para indução de uma árvore e a avaliação dos modelos induzidos;
- No capítulo 3 são apresentados os conceitos de RDD, enfatizando a docagem molecular e a dinâmica molecular. Ainda neste capítulo são apresentadas a proteína alvo, bem como os ligantes utilizados em todos os experimentos desta Tese;
- O capítulo 4 apresenta um repositório de dados chamado FReDD, desenvolvido para o armazenamento de todos os dados relacionados aos experimentos de docagem molecular, bem como os testes deste repositório com os dados do receptor e dos ligantes utilizados nesta Tese. Além disso, esse capítulo mostra como esse repositório facilita o pré-processamento dos dados, apresentando uma rotina de pré-processamento dos dados nele armazenado;
- No capítulo 5 é apresentado os experimentos com mineração de dados sobre os dados pré-processados a partir do FReDD. São realizados experimentos com regras de associação, árvores

de decisão para classificação e para regressão. Para cada uma dessas técnicas utilizadas é mostrada uma nova rotina de pré-processamento, a partir dos dados já pré-processados a partir do FReDD, atendendo aos objetivos de cada tarefa de mineração;

- O capítulo 6 introduz o algoritmo 3D-Tri, apresentando as estratégias de pré-processamento dos dados, de definição do bloco para cada átomo e de indução de uma árvore de regressão para propriedades tridimensionais;
- No capítulo 7 é apresentado o teste do algoritmo proposto;
- O capítulo 8 descreve os trabalhos relacionados à esta Tese;
- No capítulo 9 são apresentadas as conclusões desta Tese, com sugestões para trabalhos futuros;
- Por fim são apresentadas as referências bibliográficas.

2. PROCESSO DE KDD E MINERAÇÃO DE DADOS

O processo de KDD apresentado por Fayyad et al. [FAY96] é uma sequência de passos interativos e iterativos de apoio à tomada de decisão, principalmente quando da existência de grandes volumes de dados. Na visão de Han e Kamber [HAN11], o processo de KDD envolve os seguintes principais itens: (a) base de dados; (b) transformação; (c) repositório alvo; (d) pré-processamento; (e) mineração de dados; (f) padrões; e (g) conhecimento. O processo inicia-se quando da existência de um grande volume de dados que merecem ser analisados. Como esse volume de dados geralmente é proveniente de fontes heterogêneas, os mesmos passam por uma etapa de transformação para que sejam representados em um único padrão de referência. Uma vez transformados, os dados são armazenados em um repositório de dados alvo, o qual é desenvolvido baseado no contexto dos dados envolvidos. Os dados devidamente organizados no repositório alvo fornecem suporte para diferentes tipos de análise, onde a principal delas é a mineração de dados. Para que os dados sejam minerados, é importante que passem por um pré-processamento, etapa importante para que os algoritmos de mineração de dados utilizados produzam resultados melhores e mais bem acurados. Os modelos induzidos pela mineração apresentam padrões que, após analisados, podem permitir atingir o conhecimento esperado.

Embora cada etapa do processo de KDD seja suficientemente abrangente para ser tratada de forma isolada, nota-se uma forte relação de dependência entre elas. Este capítulo discorre sobre as etapas do processo de KDD, dividindo as mesmas em dois grandes grupos:

- Construção de um repositório alvo, abrangendo as etapas (a), (b), (c) e (d);
- Técnicas de mineração de dados, composta pelas etapas (e), (f) e (g).

2.1 Repositório alvo

A construção de um processo de KDD se dá, geralmente, quando há interesse em analisar distintas fontes de dados, as quais apresentam seus dados, na maioria das vezes, de forma heterogênea. A ideia de construção de um repositório alvo é para que esses dados sejam armazenados em um ambiente semanticamente consistente. Nesse sentido, a heterogeneidade dos dados de origem é tratada de forma a garantir a integração dos dados em um único formato. Planejar, modelar e construir esse repositório é uma tarefa que requer conhecimento do domínio dos dados em questão. Além disso, a forma como esse repositório é modelado também depende do contexto e das necessidades de aplicação desse repositório.

Han e Kamber [HAN11] sugerem que este repositório alvo seja construído na forma de um *data warehouse* (DW), buscando manter um histórico organizado dos registros, de modo a auxiliar a tomada de decisão. Um DW é construído de forma a satisfazer sua estrutura multidimensional [KIM09]. Um modelo analítico, que comporta essa estrutura multidimensional, possui dois tipos

de tabela: fato e dimensão. Entende-se por dimensão as perspectivas de uma base de dados que possam gerar registros referentes às características do problema modelado. Essas são tipicamente organizadas em torno de um termo central, ou tabela fato, a qual contém os atributos chave das dimensões e atributos que representam valores relevantes ao problema.

Alternativamente, o repositório pode ser modelado na forma de Entidade-Relacionamento (ER) e, mesmo assim, conter todos os dados relevantes organizados e de fácil acesso a consulta. Han e Kamber [HAN11] dizem que os dados armazenados em um modelo relacional são muito utilizados para mineração por ser o tipo de repositório mais rico em dados e com maior disponibilidade.

Tendo o repositório alvo modelado, implementado e com os dados devidamente armazenados, já é possível realizar consultas analíticas. Um exemplo, para o caso de um DW, é a manipulação de seus dados por ferramentas OLAP (*On-Line Analytical Processing*). Operações como pivoteamento de dados, *roll-up*, *drill-down* e *slice&Dice* [KIM09] são operações típicas de ambientes de processamento analítico de dados. Para o caso de um repositório ER, consultas podem ser realizadas sobre esses dados a fim de extrair algum tipo de informação. Além dessas abordagens, e seguindo as demais etapas do processo de KDD, os dados uma vez armazenados estão aptos a serem utilizados por algoritmos de mineração de dados, podendo ser pré-processados para atender às necessidades da tarefa de mineração a ser empregada.

2.1.1 Pré-Processamento

Embora os dados armazenados no repositório já possam ser minerados, é interessante que os mesmos passem por uma etapa de pré-processamento. Para Tan et al. [TAN05] técnicas de pré-processamento devem ser aplicadas para que os dados se tornem mais adequados para a mineração de dados.

Mesmo que essa seja uma etapa trabalhosa e que demanda boa parte do tempo consumido durante todo o processo de KDD, este esforço merece especial atenção por contribuir fortemente para o grau de confiança e qualidade dos resultados obtidos pelos diferentes algoritmos de mineração empregados. Dentre diferentes técnicas de pré-processamento, Tan et al. [TAN05] destacam:

- Agregação: essa técnica tem por objetivo sumarizar os dados a partir de diferentes perspectivas, através da combinação de um ou mais valores em um único objeto, reduzindo o espaço a ser minerado;
- Amostragem: nessa técnica, um conjunto de atributos são selecionados de maneira aleatória para serem analisados;
- Redução de dimensionalidade: com a redução de dimensionalidade busca-se diminuir o número de atributos a serem analisados, seguindo alguma estratégia que pode estar contida nas demais aqui listadas;

- Seleção de atributos: essa técnica visa eliminar atributos que possam ser redundantes ou irrelevantes ao objetivo da mineração, de modo com que o subconjunto de dados selecionado seja tão representativo ou até mesmo melhor do que seria o conjunto de dados original;
- Criação de atributos: técnica que consiste em criar atributos, tendo como base os dados dos atributos já existentes, de modo que, ao mesmo tempo em que o escopo é reduzido, percebe-se uma melhora na qualidade dos dados;
- Discretização: nesta técnica os atributos contínuos são transformados em intervalos que representam classes categóricas;
- Transformação de variáveis: utilizada para modificar os objetos de um determinado atributo, tendo como base uma mesma regra como, por exemplo, a normalização dos valores.

Essas técnicas podem ser combinadas entre si para que um arquivo de entrada seja adequadamente produzido e, assim, permitir obter melhores resultados com os experimentos de mineração de dados.

2.2 Mineração de dados

Mineração de dados é a etapa do processo de KDD que converte dados brutos em informação. A mineração de dados pode ser dividida em dois tipos de tarefas: descritivas e preditivas. Tarefas descritivas sumarizam relações entre dados, tendo como objetivo melhorar o conhecimento a seu respeito. Tarefas preditivas buscam apontar conclusões a respeito dos dados analisados, predizendo um dado atributo de interesse. Dentre as tarefas descritivas, pode-se citar regras de associação e agrupamento. Já algoritmos de classificação e regressão são exemplos de tarefas preditivas [TAN05].

Para o melhor entendimento do trabalho realizado nesta tese, este capítulo mostra os algoritmos para essas diferentes tarefas. Para o caso de tarefas preditivas, esse capítulo focaliza em algoritmos de indução de árvores de decisão, seja para classificação ou para regressão.

2.2.1 Regras de Associação

Algoritmos de regra de associação ocupam-se de identificar relações entre itens frequentes, onde uma regra de associação é uma implicação na forma $X \rightarrow Y$ [ALP10] onde X e Y são conjuntos não vazios. Ao minerar itens frequentes, busca-se por relações recorrentes em um mesmo conjunto de dados. Supondo que $I = \{I_1, I_2, \dots, I_n\}$ seja um conjunto de itens em T , a implicação $X \rightarrow Y$ dá-se por: $X \subset I$, $Y \subset I$ e $X \cap Y = \phi$.

Há duas medidas tipicamente utilizadas para medir o quão interessante é uma regra encontrada: suporte e confiança. Essas regras dizem respeito à utilização e à certeza da regra, respectivamente [HAN11], onde:

$$\text{Suporte}(X \rightarrow Y) = P(X \cup Y) \quad (2.1)$$

$$\text{Confiança}(X \rightarrow Y) = P(Y|X) \quad (2.2)$$

Assim, tendo um conjunto de transações T , o algoritmo busca encontrar todas as regras onde $\text{Suporte} \geq \text{SuporteMinimo}$ e $\text{Confiança} \geq \text{ConfiançaMinima}$, sendo que SuporteMinimo e ConfiançaMinima são seus respectivos limites parametrizados [TAN05]. A construção de regras de associação pode ser vista como dois principais passos:

- Geração de itens frequentes: esse passo busca encontrar todos os itens que satisfazem o limite de SuporteMinimo . A esse conjunto de itens dá-se o nome de itens frequentes;
- Geração de Regras: nessa etapa o objetivo é extrair as regras que satisfaçam o limite da ConfiançaMinima a partir dos itens frequentes encontrados no passo anterior.

O algoritmo de regra de associação mais utilizado é o Apriori, proposto por Agrawal et al. [AGR93]. Este algoritmo segue o princípio de que se um conjunto de itens é frequente, todos os subconjuntos desse item também o são.

2.2.2 Agrupamento

Algoritmos de agrupamento, ou segmentação, tem por objetivo agrupar objetos em classes com objetos similares. Nesse sentido, um agrupamento é uma coleção de objetos similares entre si, e diferentes de objetos pertencentes a um outro agrupamento [HAN11]. Trata-se de um método não supervisionado de aprendizagem, onde os exemplos não estão associados a uma categoria, ou atributo de interesse.

Uma das maneiras mais simples de medir a similaridade entre objetos é através da distância entre padrões de um objeto. Para objetos cujas características são numéricas, a distância pode ser medida pela distância Euclidiana entre dois pontos em um espaço de multidimensional. Ainda, existem métodos que fazem uso de probabilidades para medir essa similaridade. Assim, um conjunto inicial de objetos é selecionado para integrar cada agrupamento, e os demais objetos são agregados com base em um cálculo de probabilidade das características desse objeto serem similares com o grupo ao qual está-se agregando.

Dentre diferentes algoritmos de agrupamento, este capítulo concentra-se no K-means [HAR79]. Esse é um algoritmo baseado em protótipo, que tenta encontrar um número específico de grupos (k), os quais são representados por seus protótipos. Em conjuntos de dados numéricos o protótipo se dá na forma de um centróide, o qual é a média de todos os pontos do grupo. Quando o centróide não é significativo, ou seja, quando há atributos categóricos no conjunto de dados, o protótipo é definido pelo medóide. Um medóide é o ponto mais representativo no grupo. O K-medóides, como é chamado o algoritmo para o caso do protótipo ser um medóide, é comumente aplicado para conjuntos de dados científicos [TAN05]. O pseudo-código do K-means está representado no Algoritmo 2.1.

Algoritmo 2.1: Pseudo-código do K-means. Adaptado de [TAN05] e [HAR79]

- 1: selecione k pontos como sendo os centróides iniciais
 - 2: **repita**
 - 3: Compute k agrupamentos atribuindo objetos que sejam o mais próximo do seu centróide
 - 4: Recompute o centróide de cada cluster
 - 5: **até** que os centróides não mudem
-

2.2.3 Predição

Tarefas preditivas de mineração de dados buscam construir modelos que apresentem a melhor combinação de relacionamentos entre um conjunto de atributos, denominados atributos preditivos, em função de um dado atributo de interesse, ou atributo alvo [TAN05]. Em predição existem basicamente um conjunto de dados de entrada x e uma saída y , onde a tarefa é aprender como mapear o conjunto de entrada para a saída. Esse mapeamento pode ser definido como uma função $y = g(x|\theta)$ onde $g(\cdot)$ é o modelo e θ são os seus parâmetros [ALP10].

Algoritmos de predição podem ser aplicados tanto para classificação quanto para regressão. A classificação se dá quando o conjunto de dados a ser minerado possui como atributo alvo valores categóricos, ou seja, valores que podem ser separados em classes que descrevem os atributos preditivos. Já algoritmos de regressão são aplicados onde o atributo alvo de um conjunto de dados é numérico.

Ainda que existam diferentes tipos de algoritmos utilizados para predição, muitos deles apenas constroem uma função preditiva que rotula o atributo alvo dos objetos minerados. Esse é o caso de algoritmos desenvolvidos na forma de uma caixa-preta, como redes neurais e SVM. Freitas et al. [FRE10] argumenta que apesar da falta de consenso na literatura de mineração de dados a respeito de algoritmos que produzem resultados mais compreensíveis, existe um acordo de que representações na forma de árvores de decisão e conjuntos de regras podem ser melhor compreendidos por usuários finais do que representações do tipo caixa-preta.

2.2.3.1 Indução de árvores de decisão

Árvores de decisão podem ser vistas como um dos métodos mais utilizados para inferência indutiva, onde a indução é feita a partir de um conjunto de dados rotulados, ou seja, que contenham um valor como atributo alvo. Uma árvore de decisão é uma estrutura de dados hierárquica implementada a partir de uma estratégia de dividir para conquistar. A vantagem de uma árvore de decisão é que ela representa o conhecimento descoberto na forma de um grafo, sendo que sua estrutura hierárquica é capaz de apontar a importância dos atributos utilizados para predição.

Alpaydim [ALP10] explica que uma árvore de decisão é um modelo hierárquico de aprendizagem supervisionada, onde regiões locais são identificadas em sequências recursivas de divisões do conjunto de dados. Uma árvore é composta por nodos de decisão internos e por nodos terminais, os quais são chamados de folha. Cada nodo m implementa uma função de teste $f_m(x)$ com resultados discretos que servem para rotular as arestas. Assim, a partir de um conjunto de dados de entrada, um teste

é aplicado para cada nodo e uma das arestas é percorrida, dependendo do resultado de rótulo. Esse processo inicia em um nodo denominado raiz e é recursivamente repetido até atingir um nodo folha, o qual descreve a saída, ou atributo alvo. Esse é um método não-paramétrico eficiente tanto para classificação quanto para regressão.

A maioria dos algoritmos de árvores de decisão faz uso de uma estratégia gulosa para a indução da árvore, onde a partição de um nodo é feita com base em um parâmetro que identifica o ótimo local para este nodo. O algoritmo básico de indução de árvore é o algoritmo de classificação de Hunt, o qual serviu de base para a construção de algoritmos clássicos de indução de árvore de decisão, como o ID3 [QUI86], C4.5 [QUI93] e CART [BRE84]. A estratégia do algoritmo de Hunt é dada como segue:

1. Escolha um atributo;
2. Estenda a árvore adicionando um ramo para cada valor do atributo;
3. Considerando o atributo escolhido, passe os exemplos para as folhas;
4. Para cada folha:
 - (a) Se todos os exemplos pertencerem ao mesmo atributo alvo, associe este atributo à folha;
 - (b) Senão, repita os passos de 1 a 4.

A Figura 2.1 ilustra um conjunto de dados (à esquerda) e sua correspondente árvore de decisão (à direita). Esse conjunto de dados é composto por uma série de dados rotulados em C_1 , para o caso dos círculos, e C_2 , para o caso dos retângulos.

Para a indução de modelos bem acurados, algoritmos de indução de árvore devem abordar algumas questões importantes, Tan et al. [TAN05] indica as duas mais importantes como:

- Qual a melhor maneira de particionar os atributos? Cada passo recursivo do crescimento da árvore demanda a seleção de um atributo para que, a partir de um teste de condição desse atributo, o conjunto de dados seja dividido em subconjuntos. Para tanto, o algoritmo precisa implementar um método que avalie qual o melhor atributo a ser utilizado no momento, em busca de um ótimo local;
- Como o procedimento de partição dos atributos deve parar? É preciso estabelecer uma condição de parada de crescimento da árvore. Uma estratégia possível é seguir expandindo até todos os exemplos pertencerem ao mesmo atributo alvo ou todos os exemplos terem atributos de igual valores. Apesar disso, outras estratégias devem ser analisadas para que o procedimento de crescimento termine antes, sempre visando alguma vantagem em relação ao modelo.

Questões de particionamento de atributos e critério de parada devem ser implementadas de acordo com o objetivo do algoritmo e tipos de dados envolvidos.

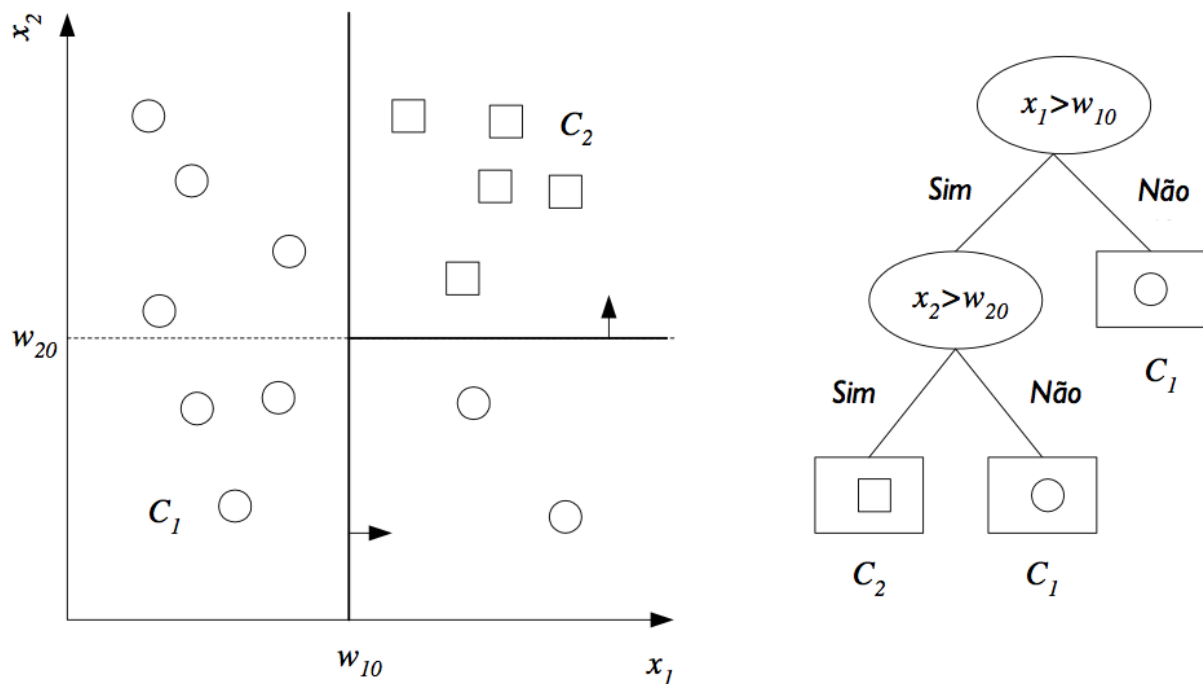


Figura 2.1: Conjunto de dados e sua respectiva árvore de decisão. Adaptado de [ALP10]

2.2.3.2 Indução de árvores de decisão para classificação

Classificação é o processo de encontrar um modelo ou função que descreva e diferencie classes de um determinado conjunto de dados, com o propósito de utilizar esse modelo para prever classes de objetos cuja categoria é desconhecida [HAN11]. Tan et al [TAN05] complementa que esta é a tarefa de aprender uma função f que mapeia cada conjunto de atributos x à uma das y classes pré-definidas. Neste caso, os dados de entrada podem ser descritos por um par de atributos (X, y) sendo que X é um vetor que representa o conjunto de atributos preditivos, onde $X = (x_1, x_2, \dots, x_n)$, e y é o rótulo da classe a qual esse exemplo pertence.

Algoritmos de indução de árvore de decisão para a classificação que usam uma estratégia gulosa têm o mesmo princípio do algoritmo de Hunt, apresentando na Seção 2.2.3.1. O algoritmo, entretanto, implementa funções que buscam responder às questões de particionamento e critério de parada.

Existem diferentes medidas para selecionar o melhor particionamento, onde tais medidas são definidas em termos da distribuição das classes em relação aos exemplos, antes do particionamento. Essa medida calcula o grau de impureza do particionamento. Isto é, um particionamento é puro se, para todas as arestas, todas as instâncias de uma mesma aresta pertencem à mesma classe. Nesse sentido, assume-se que c seja o número de classes e $p(i|t)$ seja a fração de exemplos que pertencem à classe i para um dado nodo t , busca-se encontrar o menor grau de impureza para esse particionamento, onde exemplos de medidas de impureza do nodo são Entropia [QUI86] e Gini [BRE84].

$$Entropia(t) = - \sum_{i=0}^{c-1} p(i|t) \log_2 p(i|t) \quad (2.3)$$

$$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2 \quad (2.4)$$

Dadas essas medidas, é possível calcular quão bem um dado teste ocorreu. Para tanto, é calculado o total de impureza após a partição, ou seja, o ganho de informação da partição.

$$GanhoInfo = I(nodo) - \sum_{j=1}^k \frac{N(v_j)}{N} I(v_j), \quad (2.5)$$

onde $I(.)$ é o cálculo da medida de impureza utilizada, N é o número total de exemplos, k é o número de atributos sendo utilizados e v_j é um nodo filho do nodo sendo avaliado.

O critério de parada do algoritmo é geralmente implementado na forma de um limiar do grau de pureza $I(.)$ calculado, onde esse limiar é definido como θ_I . Por esse limiar entende-se que o algoritmo não pára de induzir a árvore apenas quando a impureza seja exatamente 0 ou 1, mas quando ela está perto o suficiente, conforme limiar parametrizado.

Dadas essas medidas, um algoritmo de indução de árvore de decisão para classificação pode ser implementado conforme ilustra o Algoritmo 2.2 [ALP10].

A qualidade dos modelos induzidos podem ser avaliadas a partir de diferentes métricas. Dentre elas pode-se citar:

- Acurácia;
- Medida-F;
- Tamanho da árvore.

A acurácia de um modelo representa quão satisfatória foi a classificação. Uma das maneiras de avaliar a qualidade é fazer a indução do modelo a partir de um método denominado validação cruzada [WIT11]. Por esse método, conjunto de dados é dividido em em n partições, onde utiliza-se $n - 1$ para treino e 1 para teste, n vezes. Logo, é possível determinar as instâncias que foram preditas corretamente, onde tem-se:

- Número de instâncias classificadas como verdadeiro positivo (VP);
- Número de instâncias classificadas como verdadeiro negativo (VN);
- Número de instâncias classificadas como falso positivo (FP);
- Número de instâncias classificadas como falso negativo (FN).

Assim, a acurácia do modelo é calculada conforme Equação 2.6. O resultado da acurácia diz respeito uma taxa de acerto, a qual compreende uma faixa de 0 a 100%, onde quanto mais próximo de 100%, melhor.

 Algoritmo 2.2: Árvore de decisão para classificação, adaptado de [ALP10].

```

Procedure GeraArvore( $X$ )
1: se  $Impureza(X) < \theta_I$  então
2:   Cria folha rotulada com a maioria das classes em  $X$ 
3:   retorna
4: fim se
5:  $i \leftarrow ParticionaAtributo(X)$ 
6: para cada ramo de  $x_i$  faça
7:   Encontre  $X_i$  seguindo no ramo
8:   GeraArvore( $X_i$ )
9: fim para
Procedure ParticionaAtributo( $X$ )
10:  $MinImpureza \leftarrow MAX$ 
11: para todos atributos  $i = 1, \dots, d$  faça
12:   se  $x_i$  é discreto com  $n$  valores então
13:     Particiona  $X$  em  $X_1, \dots, X_n$  por  $x_i$ 
14:      $e \leftarrow GanhoInfo(X_1, \dots, X_n)$ 
15:     se  $e < MinImpureza$  então
16:        $MinImpureza \leftarrow e$ 
17:     senão
18:        $MelhorParticao \leftarrow i$ 
19:     fim se
20:   senão
21:     para todos partições possíveis faça
22:       Particiona  $X$  em  $X_1, X_2$  sobre  $x_i$ 
23:        $e \leftarrow GanhoInfo(X_1, X_2)$ 
24:       se  $e < MinImpureza$  então
25:          $MinImpureza \leftarrow e$ 
26:       senão
27:          $MelhorParticao \leftarrow i$ 
28:       fim se
29:     fim para
30:   fim se
31: fim para
32: retorna  $MelhorParticao$ 

```

$$Acurácia = \frac{PrediçõesCorretas}{TotaldePredições} = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.6)$$

Outra maneira de avaliar modelos de classificação é com uma métrica denominada medida-F. Essa faz uso de duas outras medidas denominadas precisão e revocação. De acordo com Han & Kamber [HAN11], essas medidas são assim calculadas:

$$Precisão = \frac{|Relevantes \cap Recuperadas|}{|Recuperadas|} \quad (2.7)$$

$$Revocação = \frac{|Relevantes \cap Recuperadas|}{|Relevantes|} \quad (2.8)$$

Onde entende-se por:

- $|Relevantes \cap Recuperadas|$: as instâncias que foram recuperadas corretamente (VP);
- $|Recuperadas|$: total de instâncias recuperadas, sejam elas corretas ou não (VP + FP);
- $|Relevantes|$: todas as instâncias que foram recuperadas corretamente, mais as instâncias que não foram recuperadas mas que deveriam ter sido (VP + FN).

A medida-F é uma composição dessas duas métricas, conforme ilustra Equação 2.9. O resultado apresenta um valor entre 0 e 1, onde os valores melhores são aqueles mais próximos de 1.

$$Medida - F = \frac{Precisão \times Revocação}{(Precisão + Revocação)/2} \quad (2.9)$$

Por fim, uma outra métrica para avaliar não a qualidade do modelo, mas a compreensibilidade do mesmo, é o tamanho da árvore. Essa diz respeito à profundidade da árvore, ou seja, quantos níveis existem até o nó folha mais profundo.

2.2.3.3 Indução de árvores de decisão para regressão

Regressão é um modelo de predição em que o atributo alvo é contínuo. Essa é a tarefa de aprender uma função alvo f que mapeia cada conjunto de atributos X para uma saída de valores contínuos y . Árvores de decisão para predição numérica são chamadas de árvores de decisão para regressão, quando a função f mapeada para cada nó folha contém a média dos valores de y para os exemplos que compõem uma folha. Além de árvores de regressão também existem as árvores que rotulam cada nó com um modelo de regressão linear, sendo essa técnica chamada de árvores modelo.

Segundo Witten et al [WIT11] e Alpaydim [ALP10], tanto árvores de regressão quanto árvores modelo seguem o mesmo princípio de indução de árvores para classificação, mudando o foco do método de particionamento e critério de parada. Um dos algoritmos que implementa árvores de regressão e árvores modelo, e cujos critérios são utilizados para esta Tese, é o M5P [QUI92] [WAN97]. O maior objetivo deste algoritmo é maximizar a redução do desvio padrão (RDP), considerando o desvio padrão dos exemplos no conjunto de dados $dp(X)$.

$$RDP = dp(X) - \sum_i \frac{|X_i|}{|X|} \times dp(X_i) \quad (2.10)$$

É pelo valor de RDP que o algoritmo particiona os atributos: para cada ciclo recursivo da indução, o atributo escolhido é aquele que apresenta o maior valor de RDP . Com relação ao critério de parada, este considera limites do número de exemplos em X , bem como um limiar em relação a uma taxa do $dp(X)$. O pseudo-código adaptado do M5P está descrito no Algoritmo 2.3.

Algoritmo 2.3: Árvore de decisão para regressão, adaptado de [ALP10] e [WAN97].

Procedure GeraArvore(X)

1: **se** DP não foi calculado **então**
 2: $DP \leftarrow dp(X)$
 3: **fim se**
 4: **se** Número de exemplos em $X < 4$ ou $dp(X) < 0.05 \times DP$ **então**
 5: Cria folha rotulada com a média dos valores de y em X
 6: **retorna**
 7: **fim se**
 8: $i \leftarrow \text{ParticionaAtributo}(X)$
 9: **para** cada ramo de x_i **faça**
 10: Encontre X_i seguindo no ramo
 11: GeraArvore(X_i)
 12: **fim para**

Procedure ParticionaAtributo(X)

13: **para todos** atributos $i = 1, \dots, d$ **faça**
 14: Calcula RDP
 15: **se** x_i é discreto com n valores **então**
 16: Particiona X em X_1, \dots, X_n por x_i
 17: $\text{MelhorParticao} \leftarrow i$ com maior valor de RDP
 18: **senão**
 19: **para todos** partições possíveis **faça**
 20: Particiona X em X_1, X_2 sobre x_i
 21: $\text{MelhorParticao} \leftarrow i$ com maior valor de RDP
 22: **fim para**
 23: **fim se**
 24: **fim para**
 25: **retorna** MelhorParticao

O objetivo da regressão é induzir um modelo tal que a sua função f minimize um erro. Típicas funções de erro para tarefas de regressão são Erro Médio Absoluto (MAE - *Mean Absolute Error*) e Erro Médio Quadrático (RMSE - *Root Mean Squared Error*). Essas duas medidas são calculadas conforme segue, onde p é o valor predito e a é o valor real. Essas duas medidas retornam valores em uma faixa de 0 a 1, sendo que quanto mais próximo de 0, melhor o resultado.

$$MAE = \frac{|(p_1 - a_1)| + \dots + |(p_n - a_n)|}{n} \quad (2.11)$$

$$RMSE = \sqrt{\frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{n}} \quad (2.12)$$

Ainda, é possível medir a correlação estatística entre a e p , cujo cálculo está apresentado na Equação 2.13. Esses valores variam entre 0 e 1, onde 1 apresenta uma correlação perfeita e 0 a ausência de correlação. Além disso, o valor -1 também é válido, o qual apresenta uma correlação inversa perfeita.

$$\text{Correlação} = \frac{S_{PA}}{\sqrt{S_P S_A}} \quad (2.13)$$

onde $S_{PA} = \frac{\sum_i (p_i - \bar{p})(a_i - \bar{a})}{n-1}$, $S_P = \frac{\sum_i (p_i - \bar{p})^2}{n-1}$, e $S_A = \frac{\sum_i (a_i - \bar{a})^2}{n-1}$

2.3 Considerações do capítulo

Este capítulo discorreu sobre o processo de KDD, enfatizando as etapas de construção de um repositório alvo, de pré-processamento e de mineração de dados. Com relação à construção de um repositório foi relatado que esse pode ser construído tanto na forma de um DW como na forma de um modelo relacional. Esse repositório serve de base para que diferentes técnicas de pré-processamento sejam aplicadas para atender a algum objetivo de mineração de dados. Com relação à etapa de mineração de dados, apresentou-se os conceitos de algoritmos descritivos e preditivos, onde o primeiro sumariza dados e apresenta relações entre eles, e o segundo constrói um modelo que apresente a melhor combinação de relacionamentos entre atributos preditivos em relação a um atributo alvo. Para tarefas descritivas apresentou-se os algoritmos de regras de associação e algoritmos de agrupamento. Já para tarefas preditivas enfatizou-se árvore de decisão, explicando como as mesmas são geradas tanto para classificação como para regressão.

3. DOCAGEM MOLECULAR CONSIDERANDO O RECEPTOR FLEXÍVEL

Inicialmente a bioinformática era definida como uma área interdisciplinar envolvendo biologia, ciência da computação, matemática e estatística para analisar dados biológicos. Com o advento da era genômica, bioinformática passou a ser definida em termos de moléculas e a aplicação da computação para entender e organizar informação associada a esses dados biológicos em larga escala [LUS01] [MOU04].

Para Lesk [LES02], uma das principais características dos dados de bioinformática é o seu grande volume. Bancos de dados biológicos não são apenas extensos, mas crescem a uma taxa bastante elevada. Esse grande volume de dados e seu crescimento definem os objetivos da bioinformática: organizar dados biológicos de maneira que os pesquisadores possam acessar informações já cadastradas, assim como submeter novas entradas, na medida em que vão sendo produzidas. Além disso, a bioinformática também ocupa-se em realizar pesquisas e desenvolver ferramentas que ajudem na análise desses dados para interpretar os dados de forma que tenham significado biológico [LUS01]. Dentre as diversas áreas de atuação em bioinformática, Lesk [LES02] aponta algumas frentes como: genômica, proteômica, alinhamentos de árvores filogenéticas, biologia de sistemas e descoberta de fármacos.

Esta Tese está inserida no contexto de Desenho Racional de Fármacos (RDD - *Rational Drug Design*) [KUN92]. O princípio fundamental do RDD é a interação entre receptores e ligantes [LYB95]. Ligantes são definidos como moléculas que se ligam a outras moléculas biológicas, chamadas receptores, para realizar ou inibir funções específicas [BAL09]. É na docagem molecular que se investiga e avalia o melhor encaixe do ligante no receptor [KUN92]. Um dos maiores desafios dessa área de pesquisa é lidar com o grande volume de dados envolvidos, como catalogação de ligantes, conformações do receptor obtidas por simulações pela dinâmica molecular (DM) e resultados de experimentos de docagem molecular. Para a execução de docagem molecular é necessário um receptor, um ligante e um software para executar as simulações. Nesse trabalho considera-se como receptor a enzima InhA do *Mycobacterium tuberculosis* (Mtb) [DES95]; como ligantes, NADH [DES95], TCL [KUO03], PIF [OLI04] e ETH [WAN07]; e como software de docagem o AutoDock3.0.5 [MOR98].

Este capítulo discorre sobre RDD, em especial sobre os experimentos de docagem molecular e aquisição de dados com as moléculas sendo utilizadas.

3.1 Desenho racional de fármacos

A indústria farmacêutica encontra-se constantemente sob pressão para aumentar a taxa com que novos medicamentos são inseridos no mercado [LYN02]. Hoje em dia, o tempo para que um novo fármaco seja disponibilizado para comercialização é de 10 a 15 anos, e os custos associados são estimados em 1,2 bilhões de dólares [CAS07] [KAP08]. Por essas razões, existem diversos esforços

sendo aplicados para tentar reduzir tanto o tempo como o custo e, ao mesmo tempo, aumentar a qualidade dos compostos candidatos a fármacos.

Avanços em biologia molecular, modelagem computacional e ferramentas de simulações tem apresentado um impacto positivo no processo de planejamento de fármacos, tornando viável a aplicação de RDD. A abordagem *in-silico* para o RDD é um processo que combina informação estrutural e esforços computacionais [KUN92] baseados no entendimento da interação entre uma proteína alvo, ou receptor, e diferentes ligantes, ou pequenas moléculas. O RDD é um ciclo que combina quatro etapas:

1. O primeiro passo consiste em isolar um alvo específico, ou receptor. A partir de análises computacionais a respeito da sua estrutura tridimensional (3D), a qual é armazenada em um banco de dados estrutural, como o PDB (*Protein Data Bank*) [BER00], é possível identificar regiões de ligações como, por exemplo, regiões onde uma pequena molécula pode se ligar a esse receptor;
2. Com base na provável região de ligação identificada no passo anterior, é selecionado um conjunto de ligantes candidatos a se ligarem nesse receptor. As diferentes conformações que um dado ligante pode assumir dentro do sítio ativo de uma proteína em particular, pode ser simulado por software de docagem molecular, como o Autodock3.0.5 [MOR98];
3. Ligantes que teoricamente obtiverem os melhores resultados em simulações, são experimentalmente sintetizados e testados;
4. Baseado nos resultados experimentais, um novo medicamento pode ser gerado, ou o processo volta ao passo 1.

3.1.1 Docagem molecular e simulações por dinâmica molecular

A interação entre moléculas é o princípio do desenho de fármacos. É na docagem molecular que se investiga e avalia o melhor encaixe do ligante na estrutura alvo ou receptor. De acordo com Lybrand [LYB95], um ligante deve interagir com um receptor para exercer uma função fisiológica vinculada à ligação dessa com outras moléculas, e essas ligações determinam se as funções do receptor serão estimuladas ou inibidas. Essas ligações ocorrem em locais específicos, chamados sítios ativos ou de ligação. A associação entre duas moléculas no sítio de ligação não depende somente do encaixe. Existe a necessidade de haver uma energia favorável para que essa interação ocorra. Essa energia é determinada pela carga e tamanho dos átomos nele contidos. Essas ligações que ocorrem entre os átomos são medidas pela quantidade de energia despendida, sendo que quanto mais negativa, melhor a interação entre as moléculas. Conforme Jeffrey [JEF97], a maior distância que permite um contato significativo entre átomos do receptor e do ligante, para que haja energia liberada, é de 4.0 Å.

Uma das maneiras de avaliar um resultado de docagem é através do valor estimado da energia livre de ligação (FEB - *Free Energy of Binding*). O Autodock é um software utilizado para prever a

ligação de uma conformação de um ligante em um receptor, aplicando uma técnica que combina um algoritmo de busca de conformações com um método baseado em grade para avaliação da energia [GOO96]. Esse método baseado em grade é executado pelo módulo AutoGrid do Autodock3.0.5. Esse módulo pré-calcula uma grade tridimensional de energias de interações, tendo como base as coordenadas do receptor e do ligante. Um exemplo de grade utilizada neste trabalho está descrito na Figura 3.1, considerando o receptor InhA e o ligante PIF.

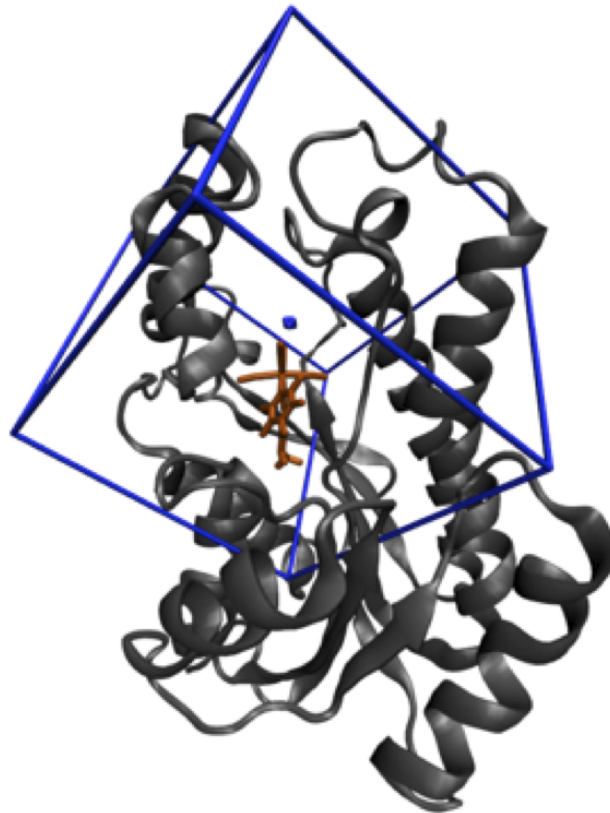


Figura 3.1: Grade 3D considerando o receptor InhA e o ligante PIF.

A maioria dos algoritmos que executam docagem molecular somente considera a flexibilidade do ligante, considerando o receptor rígido. Entretanto, sabe-se que as proteínas não permanecem rígidas em seu ambiente celular, sendo de fundamental importância a consideração dessa flexibilidade do receptor na execução dos experimentos de docagem molecular.

Existem diferentes trabalhos que visam a incorporação da flexibilidade de receptores na docagem molecular, como revisado por Totrov e Abagyan [TOT08] e Rao et al. [RAO09]. Dentre as diferentes abordagens, nesta Tese utiliza-se a execução de uma série de experimentos de docagem molecular, considerando em cada experimento uma conformação do receptor gerada por uma simulação de dinâmica molecular (DM) [LIN02] [MAC07] [AMA08]. A simulação por DM é uma das técnicas computacionais mais versáteis e amplamente utilizadas para o estudo de macromoléculas biológicas [van90]. Com simulações pela DM é possível estudar o efeito explícito de ligantes na estrutura e estabilidade das proteínas, os diferentes parâmetros termodinâmicos envolvidos, incluindo energias de interação e entropias.

3.2 Aquisição de dados

3.2.1 Receptor

O receptor sendo utilizado nesta Tese é a enzima InhA do *Mycobacterium tuberculosis* (MTB) [DES95]. Essa enzima, a qual pode ser vista como um importante alvo para o controle da tuberculose [OLI07], contém um total de 268 resíduos de amino-ácido, totalizando 4.008 átomos. Nesse trabalho o receptor é totalmente flexível e sua representação foi obtida a partir de uma trajetória de simulação por dinâmica molecular coletada por 3.100 ps (onde um ps corresponde a 10^{-12} segundos), conforme descrito em [SCH05]. Cada 1.0 ps da simulação por dinâmica molecular corresponde a uma conformação em um formato PDB [BER00]. Parte deste arquivo é apresentado na Figura 3.2, onde cada registro representa as coordenadas de um átomo do receptor. A segunda e terceira coluna correspondem ao número e nome do átomo, respectivamente. O nome e o número do resíduo ao qual este átomo pertence, aparecem na quarta e quinta coluna. As colunas seis, sete e oito correspondem às coordenadas x, y, z do átomo. Essa figura mostra os 12 primeiros átomos que fazem parte do primeiro resíduo (ALA1) da InhA.

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	4008	OXT	LEU	268	-20.647	-17.857	-3.495	0.00	0.00

Figura 3.2: Parte do arquivo PDB, que corresponde a 1.0 ps da trajetória de simulação por DM da enzima InhA.

Esse conjunto de conformações é utilizado para representar a flexibilidade explícita do receptor InhA durante o procedimento de experimentos de docagem molecular considerando o receptor flexível [MAC07]. A esse modelo totalmente flexível dá-se o nome de modelo FFR (*Fully-Flexible Receptor*).

Para ilustrar a flexibilidade desse receptor, parte da estrutura 3D da enzima InhA pode ser visualizada na Figura 3.3, onde cada cor representa uma conformação distinta. A estrutura cristalográfica (PDB ID: 1ENY) obtida do Protein Data Bank (PDB) [BER00] está representada em laranja; as outras quatro estruturas são conformações médias que variam de 0.0 a 500 ps (em ciano), de 500 a 1.000 ps (em azul), de 1.050 a 1.500 ps (em magenta) e de 1.550 a 2.000 ps (em verde).

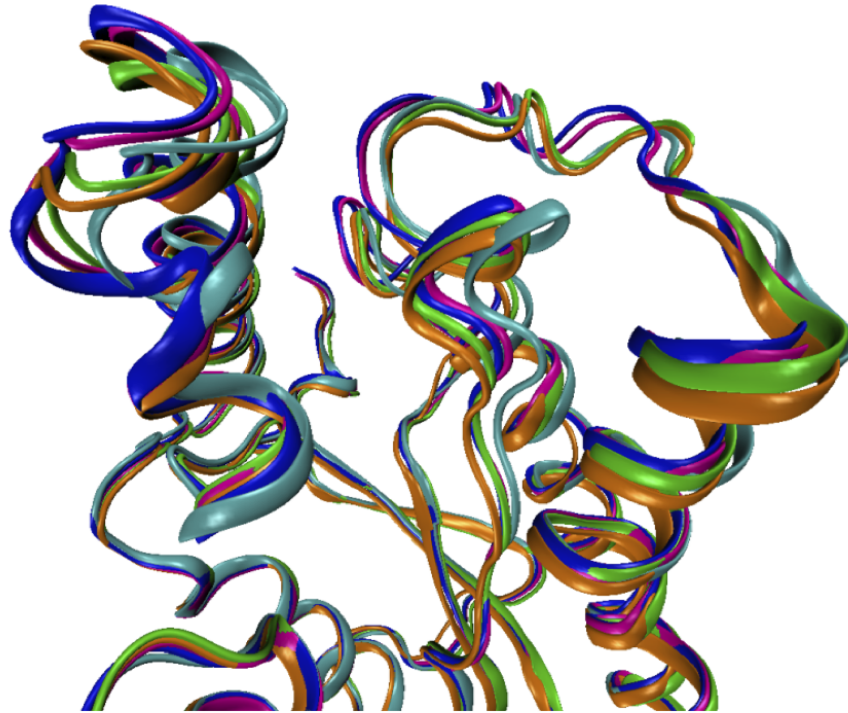


Figura 3.3: Parte da conformação 3D do modelo FFR da enzima InhA. Cada cor representa uma conformação distinta. A estrutura cristalográfica (PDB ID: 1ENY) obtida do Protein Data Bank (PDB) [BER00] está representada em laranja; as outras quatro estruturas são conformações médias que variam de 0.0 a 500 ps (em ciano), de 500 a 1.000 ps (em azul), de 1.050 a 1.500 ps (em magenta) e de 1.550 a 2.000 ps (em verde).

3.2.2 Ligantes

Estão sendo considerados os ligantes NADH, TCL, PIF e ETH, cuja informações atômicas estão listadas na Tabela 3.1. Além disso, a estrutura tridimensional de cada um deles está ilustrada na Figura 3.4, na forma de palitos, sendo que o NADH aparece na Figura 3.4(a), o PIF na Figura 3.4 (b), o TCL na Figura 3.4 (c) e o ETH na Figura 3.4 (d).

Tabela 3.1: Nomes, abreviaturas e número de átomos dos ligantes sendo utilizados

Nome	Abreviatura	Número de Átomos
Nicotinamida adenina dinucleotídeo	NADH	52
Triclosano	TCL	18
Isoniazida Pentacianoferrato	PIF	24
Etionamida	ETH	13

Os arquivos dos ligantes são representados em um formato MOL2, conforme Figura 3.5. Esse formato é composto pelas seções MOLECULE, ATOM, BOND, SUBSTRUCTURE e SET. Para fins de ilustração, a Figura 3.5 mostra a seção ATOM, a qual contém informações a respeito do nome dos átomos, tipo, coordenadas x, y, z e cargas parciais.

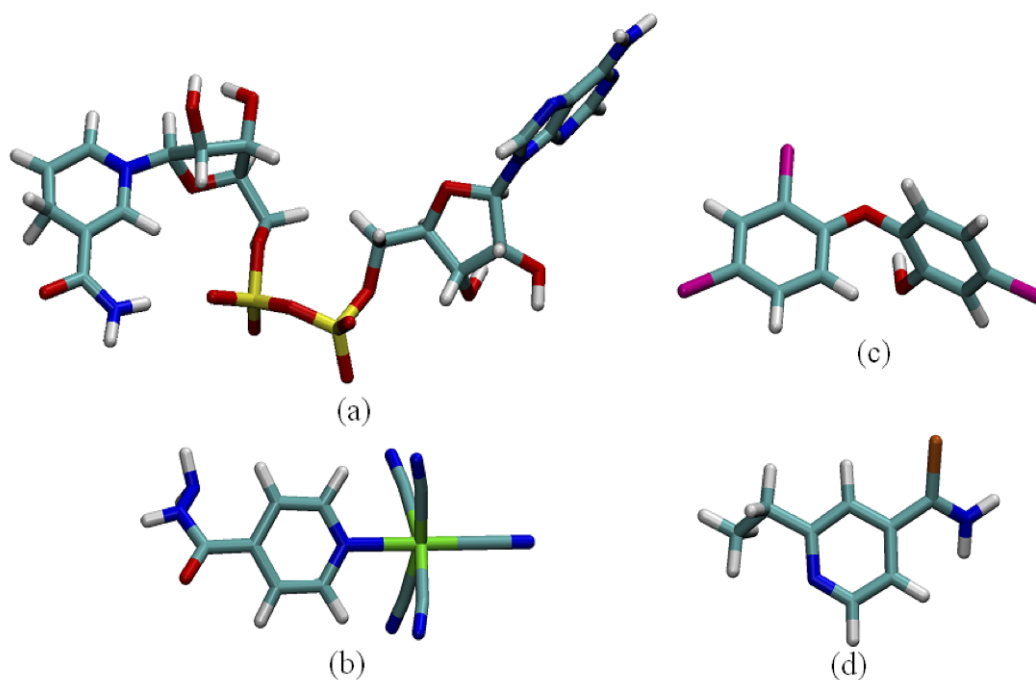


Figura 3.4: Representações das estruturas 3D dos ligantes NADH (a), PIF (b), TCL (c) e ETH (d).

```

@<TRIPOS>MOLECULE
UNTITLED
71 75 1 0 0
SMALL
USER_CHARGES

@<TRIPOS>ATOM
1 C6N      -5.5959   -4.3594    1.0216  C.2    1    ****   -0.3550
2 H6N      -5.0019   -4.4531    0.1243  H       1    ****    0.2220
3 C5N      -6.1594   -5.4464    1.7507  C.2    1    ****   -0.1730
4 H5N      -6.0147   -6.4642    1.4173  H       1    ****    0.1260
5 C4N      -6.9471   -5.1565    2.8972  C.3    1    ****    0.1350
6 H41      -7.9253   -5.5901    2.6881  H       1    ****    0.0200

...

@<TRIPOS>BOND
...
64 62 61 1
65 63 62 ar
66 64 63 1
67 65 64 1
68 66 64 1
69 67 63 ar
70 68 67 ar
71 69 68 1
72 70 68 ar
73 71 58 1
74 71 62 ar
75 71 70 ar

@<TRIPOS>SUBSTRUCTURE

```

Figura 3.5: Parte do arquivo MOL2 para o ligante NADH.

3.2.3 Experimentos de docagem molecular

Os experimentos de docagem molecular foram executados utilizando-se o workflow científico descrito em Machado et al. [MAC07]. Cada um dos quatro ligantes foram submetidos a 3.100 experimentos de docagem considerando-se, em cada experimento, uma diferente conformação do

receptor. O workflow utiliza o AutoDock3.0.5 [MOR98] como software de docagem, o qual faz uso do protocolo de *Simulated Annealing*, com 10 execuções para cada conformação. O resultado do AutoDock é um arquivo texto, conforme mostra a Figura 3.6. O resultado de cada execução é composto essencialmente pelos valores destacados nas caixas (a), (b) e (c) da Figura 3.6. A Figura 3.6(a) contém o desvio médio quadrático (RMSD), o qual indica quão distante a posição final do ligante está em relação a sua posição inicial, tipicamente estipulada por um especialista de domínio. Figura 3.6(b) contém o valor estimado de FEB e uma constante de inibição (k_i). A Figura 3.6(c) mostra as coordenadas finais dos átomos do ligante.

```

USER Run = 9
USER Cluster Rank = 1
USER Number of conformations in this cluster = 7
USER
USER RMSD from reference structure = 8.263 Å (a)
USER
USER Estimated Free Energy of Binding = -8.22 kcal/mol [(1)+(3)] (b)
USER Estimated Inhibition Constant,  $K_i$  = +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
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

Figura 3.6: Parte do resultado do AutoDock3.0.5, cujo experimento considerou uma conformação da enzima InhA e o ligante TCL.

O principal problema com a utilização da trajetória da DM em experimentos de docagem molecular é o tempo necessário para a execução de experimentos e a grande quantidade de dados gerados. Visando reduzir esse tempo de execução e melhor entender como ocorre a interação receptor-ligante considerando a flexibilidade do receptor, é necessário investir em técnicas que contribuam para esses objetivos. Para tanto, durante o desenvolvimento desta Tese aplicou-se diferentes etapas do processo de KDD, as quais são descritas nos capítulos 4 e 5 subsequentes.

3.3 Considerações do capítulo

Neste capítulo foram apresentados os conceitos de desenho racional de fármacos e de experimentos de docagem molecular. Nesta Tese os experimentos de docagem molecular fizeram uso da enzima InhA do *Mycobacterium Tuberculosis*, um importante alvo para a Isoniazida, principal fármaco para a tuberculose. Essa proteína possui uma grande flexibilidade, de modo com que sua flexibilidade explícita foi obtida através de simulações por dinâmica molecular. A esse modelo totalmente flexível

para a enzima InhA dá-se o nome de modelo FFR. Os experimentos de docagem molecular para o modelo FFR utilizaram 4 ligantes distintos: NADH, PIF, TCL e ETH. Experimentos de docagem molecular que consideram a flexibilidade explícita do receptor costumam estar relacionados com uma grande quantidade de dados, principalmente no que diz respeito às conformações tanto dos ligantes quanto do receptor. Tendo esses resultados de docagem, o desafio está em contribuir para acelerar o processo de execução de novos experimentos de docagem, desenvolvendo estratégias para identificar características de conformações que possam contribuir para essa seleção e, assim, reduzir o número de conformações a serem testadas.

4. REPOSITÓRIO ALVO PARA PRÉ-PROCESSAMENTO

Uma análise detalhada dos resultados de experimentos do modelo FFR em docagem molecular, em especial aqueles relacionados aos detalhes das interações entre ligante e receptor, é essencial para o melhor entendimento e aprimoramento do processo de experimentos de docagem como um todo. Esse tipo de análise pode ser melhor explorada se houver uma integração entre dados de simulação por DM e dos dados de docagem sobre o modelo FFR.

Considerando o grande volume de dados envolvidos, aspectos de persistência, fácil acesso e recuperação dos dados merecem ser explorados. Por essa razão foi desenvolvido um repositório abrangente para armazenar todas as características envolvidas quanto às conformações obtidas por simulações por DM bem como às relacionadas ao modelo FFR e todos seus níveis de detalhe como, por exemplo, as coordenadas espaciais dos átomos envolvidas em cada resultado de experimentos de docagem molecular. Esse repositório, denominado FReDD (*Flexible-Receptor Docking Database*), pode ser visto como uma infraestrutura eficaz para preparação de dados.

Este capítulo apresenta o repositório FReDD, o qual foi desenvolvido para integrar esses dados em todos os seus níveis de detalhe, permitindo com que análises a respeito da flexibilidade do receptor fossem realizadas. Além disso, este capítulo também mostra como o modelo do FReDD facilita a etapa de pré-processamento dos dados para a mineração, apresentando as técnicas utilizadas. Como resultados dos tópicos descritos neste capítulo, os seguintes trabalhos foram publicados durante o desenvolvimento desta Tese:

- O modelo de dados inicial do FReDD está na forma de um resumo expandido no *Brazilian Symposium on Bioinformatics* em 2009 [WIN09];
- O modelo final deste repositório, bem como análises a respeito da flexibilidade do receptor estão publicados como artigo completo no *IADIS International Conference on Applied Computing* em 2010 [WIN10a];
- O pré-processamento dos dados contidos no repositório FReDD, o qual foi utilizado para os experimentos de mineração de dados realizados durante o desenvolvimento desta Tese, foi publicado como um resumo no *ISCB Latin America* [WIN10c]. A versão completa deste pré-processamento está no artigo submetido ao *International Journal of Data Mining and Bioinformatics*, atualmente em revisão [WIN11];
- De forma resumida, tanto o modelo do repositório quanto a estratégia de pré-processamento realizada a partir do mesmo estão descritas como um capítulo do livro *Tópicos em sistemas colaborativos, multimídia, web e banco de dados* de 2010, o qual foi apresentado como minicurso durante o *Simpósio Brasileiro de Banco de Dados* em 2010. [WIN10b].

4.1 O repositório FReDD

O repositório FReDD foi desenvolvido para integrar dados de simulações por DM e resultados de experimentos de docagem molecular considerando o modelo FFR, e tendo como objetivo proporcionar um ambiente adequado para o pré-processamento desses dados. Neste sentido, este repositório é suficientemente abrangente para armazenar dados de proveniência de simulações por DM e suas respectivas conformações, integrados com as características dos resultados de docagem. Em outras palavras, o repositório FReDD permite recuperar características espaciais e temporais tanto de receptores quanto de ligantes armazenados, desde que tais conformações estejam descritas em termos de tempo de execução (para simulações de DM) e em termos de coordenadas espaciais para cada um de seus átomos.

Uma das maiores vantagens deste repositório está justamente na possibilidade de cada estrutura, seja receptor ou ligante, ser analisada em termos de seus átomos, onde cada átomo está relacionado com o resíduo com o qual o mesmo faz parte, ainda que essas características sejam armazenadas em tabelas distintas. Dessa forma, é possível executar consultas utilizando tanto abordagens *top-down* e *bottom-up*, retornando informações a respeito de ligantes e receptores em diferentes pontos de vista. O modelo do FReDD, o qual contém um total de 17 tabelas, está ilustrado na Figura 4.1. As principais características deste modelo está sumarizado na Tabela 4.1

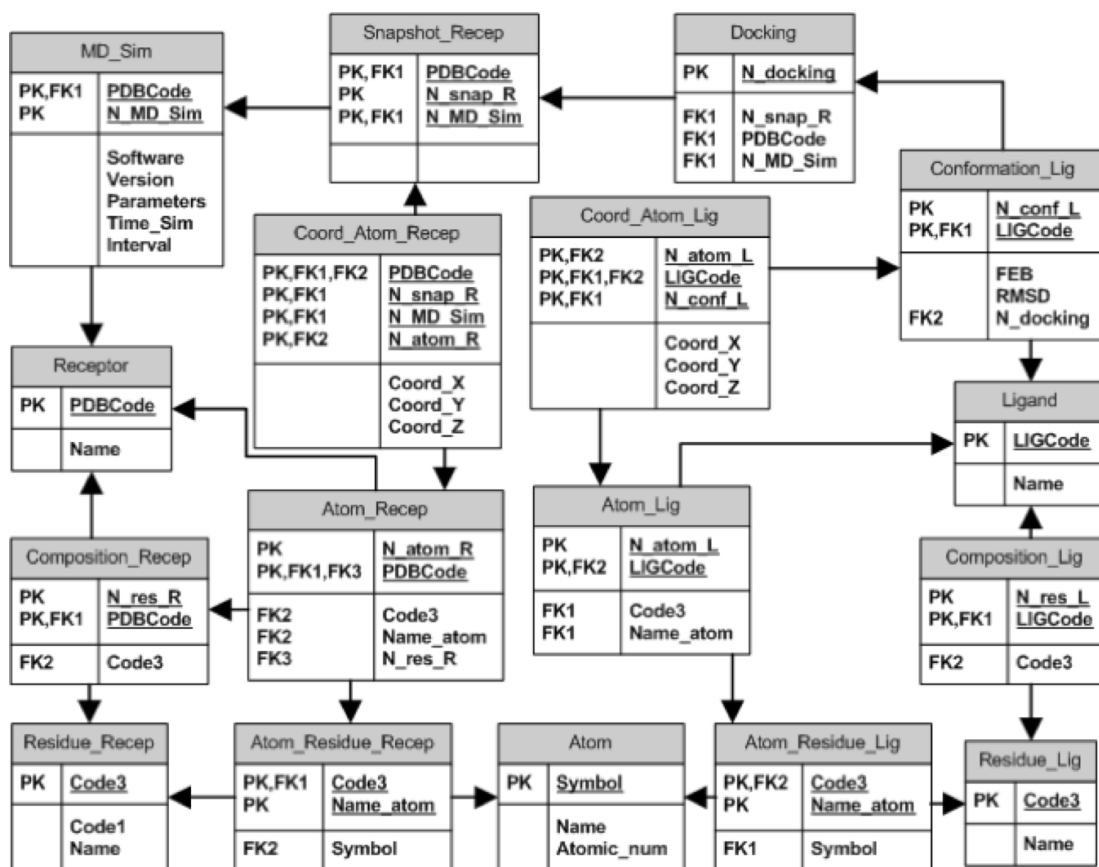


Figura 4.1: Modelo de dados do repositório FReDD.

Tabela 4.1: Descrição dos dados contidos nas tabelas do repositório FReDD

Nome da Tabela	Descrição do Conteúdo
Atom	Todos os átomos da tabela periódica
Atom_Lig	Todos os átomos de um ligante
Atom_Recep	Todos os átomos de um receptor
Atom_Residue_Lig	Relações entre os resíduos de um ligante e seus respectivos átomos
Atom_Residue_Recep	Relações entre os resíduos de um receptor e seus respectivos átomos
Composition_Lig	Relações entre um ligante e seus respectivos resíduos
Composition_Recep	Relação entre resíduos de um receptor e os 20 aminoácidos naturais
Conformation_Lig	Cada uma das execuções de um experimento de docagem molecular
Coord_Atom_Lig	Coordenadas espaciais dos átomos de um ligante
Coord_Atom_Recep	Coordenadas espaciais de uma conformação do receptor
Docking	Características da execução do experimento de docagem molecular
Ligand	Detalhes a respeito do nome e número de átomos de um ligante
MD_Sim	Dados de proveniência da simulação por DM
Receptor	Detalhes do receptor, como o cabeçalho do arquivo PDB
Residue_Lig	Todos os resíduos de um ligante
Residue_Recep	Todos os resíduos de um receptor
Snapshot_Recep	Detalhes de uma conformação, como ordem de uma trajetória de DM

4.1.1 População do repositório

Para testar o modelo de dados da Figura 4.1, está sendo considerado um receptor e quatro ligantes (os mesmos descritos no capítulo 3). Entretanto, é importante salientar que, conforme modelo da Figura 4.1, este repositório está apto a armazenar tantos receptores e ligantes quantos necessários.

A atual população de algumas das tabelas do FReDD está sumarizada na Tabela 4.2. Esta tabela trás na primeira linha informações sobre o receptor e nas demais sobre os ligantes. Na segunda coluna é mostrado, para cada ligante, o número de docagens válidas, ou seja, os experimentos de docagem que convergiram, de um total de 3.100. A terceira coluna mostra o número total de átomos para estas estruturas. Na quarta coluna são apresentadas o número total de conformações para cada estrutura, sendo que para o caso dos ligantes considera-se cada uma das 10 execuções para cada experimento de docagem. A quinta coluna indica o total de coordenadas, a qual corresponde ao produto do número de átomos pelo número de conformações.

Tabela 4.2: População do repositório FReDD

Receptor/Ligante	Docagens Válidas	Num. Átomos	Num. Conformações	Total Coordenadas
InhA		4.008	3.100	12.424.800
NADH	3.100	52	31.000	1.612.000
TCL	2.837	18	28.370	510.660
PIF	3.042	24	30.420	730.080
ETH	3.043	13	30.430	395.590

Nota-se que para o receptor, o qual possui um total de 4.008 átomos distribuídos entre seus 268 resíduos, há um total de mais de 12 milhões registros de coordenadas para a trajetória de DM sendo considerada. Com relação aos ligantes, somando o total de coordenadas para cada uma dessas estruturas, há um total de mais de 3 milhões de registros de coordenadas.

4.2 FReDD como uma infraestrutura para pré-processamento

A partir do FReDD é possível extrair diferentes tipos de informações a respeito dos dados nele armazenados. O objetivo em analisar dados de simulações de DM e resultados de experimentos de docagem é reduzir o número de conformações a serem consideradas em experimentos de docagem molecular para o modelo FFR e um dado ligante. Para tanto busca-se extrair padrões relacionados à interação ligante-receptor, os quais apresentem informações sobre quais características são importantes para serem observadas para selecionar um subconjunto de conformações. No FReDD estão disponíveis várias características que podem ser consideradas. Escolher qual delas pode ser a mais importante reflete diretamente nos experimentos de mineração a serem realizados. Aqui é apresentada uma abordagem de pré-processamento, baseado no contexto de docagem molecular, para produzir um conjunto de dados para análise.

4.2.1 Escolha do atributo alvo

Uma das maneiras de avaliar a qualidade de um experimento de docagem molecular é pelo valor estimado de FEB resultante, sendo que quanto mais negativo esse valor, melhor. Nesse sentido, o valor estimado de FEB é utilizado como o atributo alvo dos experimentos de mineração sobre os dados de docagem.

Não existe um consenso a respeito de um valor ideal para o FEB. É necessário que ele seja considerado e avaliado individualmente para cada tipo de ligante sendo utilizado. A Tabela 4.3 mostra a variação de valores de FEB, de acordo com os dados armazenados na tabela *Conformation_Lig*, para cada um dos ligantes. As colunas 2, 3 e 4 mostram os valores mínimos, máximos e a média de FEB, respectivamente e para cada ligante, para todas as 10 execuções de docagem molecular para cada conformação. As colunas 5, 6 e 7 apresentam os valores mínimos, máximos, e média de FEB, respectivamente e para cada ligante, apenas da execução que apresentou melhor valor de FEB, dentre as 10 execuções.

Tabela 4.3: Distribuição do valor de FEB para cada ligante

Ligante	10 Execuções			Melhores FEB		
	Min FEB	MaxFEB	Média	Min FEB	MaxFEB	Média
NADH	-20,61	-0,02	-9,23 ± 4,54	-20,60	0,00	-12,90 ± 4,2
PIF	-11,22	-0,01	-9,09 ± 1,63	-11,20	0,00	-9,90 ± 0,60
TCL	-10,01	-0,73	-8,17 ± 1,28	-10,00	-4,90	-8,90 ± 0,30
ETH	-8,22	-5,27	-6.63 ± 0,34	-8,22	-5,90	-6,80 ± 0,30

Nota-se, pela Tabela 4.3, que os valores de FEB para os quatro ligantes sendo utilizados são bastante distintos. Como exemplo, para o ligante NADH, nas 10, execuções o valor mínimo de FEB é -20.61 enquanto para o ligante ETH o valor mínimo é -8.22. Isso ratifica que o valor de FEB deve ser analisado por tipos de ligante pois, para o exemplo dos ligantes NADH e ETH, o melhor valor de FEB encontrado para o ligante ETH corresponde à média dos valores de FEB para o NADH.

4.2.2 Escolha dos atributos preditivos

Uma característica que contribui para a determinação do valor de FEB é a distância entre átomos dos resíduos do receptor e do ligante, onde essa distância é medida em Angstroms (Å). Isso é, para cada resíduo (*Residue_Recep*) de um receptor *R* é calculada a distância Euclidiana entre os seus átomos (*Atom_Residue_Recep*) e os átomos de um ligante *L* (*Atom_Residue_Lig*), conforme equação 4.1. x_R , y_R e z_R correspondem às coordenadas espaciais dos átomos dos resíduos do receptor (*Coord_Atom_Recep*), e x_L , y_L e z_L correspondem às coordenadas espaciais dos átomos do ligante (*Coord_Atom_Lig*).

$$Dist_{R,L} = \sqrt{(x_R - x_L)^2 + (y_R - y_L)^2 + (z_R - z_L)^2} \quad (4.1)$$

Para recuperar essas distâncias é necessário combinar todos os registros de coordenadas do receptor com todos os registros de coordenadas de cada ligante. Tendo todas as distâncias recuperadas, para formar o arquivo de entrada optou-se por fazer uso da menor distância entre os átomos de um dado resíduo do receptor e dos átomos do ligante $\min(Dist_{R,L})$, para cada resultado de experimento de docagem molecular, sendo esses os atributos preditivos. A Figura 4.2 ilustra esse conceito, onde são mostradas algumas distâncias entre o ligante PIF (em preto) e o resíduo GLY95 do receptor (em cinza). Para todas as distâncias calculadas, considera-se apenas a distância mínima que, neste exemplo, é 2,72 Å.

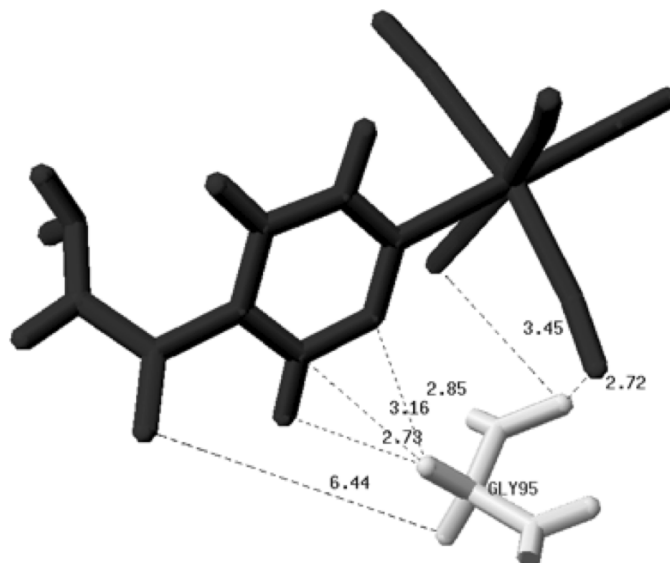


Figura 4.2: Algumas distâncias atômicas entre o ligante PIF e o resíduo GLY95 do receptor InhA.

4.2.3 Geração do arquivo de entrada

Cada resíduo do receptor é um atributo no arquivo de entrada. Como o receptor InhA contém 268 resíduos de aminoácidos, cada arquivo de entrada preparado para a mineração conta com um total de 268 atributos preditivos, mais o atributo alvo com o valor de FEB. Para cada ligante é produzido um arquivo de entrada distinto. Cada linha do arquivo de entrada é uma conformação do receptor, sendo que para cada atributo preditivo (ou resíduo) o valor no arquivo será o da menor distância entre o referido resíduo e o ligante para o qual o arquivo está sendo composto.

O Algoritmo 4.1 ilustra esse processo, onde o mesmo é aplicado para cada um dos ligantes, produzindo um arquivo de entrada para cada um deles: a matriz $[Input]$.

Algoritmo 4.1: Geração do conjunto de dados inicial.

```

1: Seja  $R$  um Receptor
2: Seja  $L$  um Ligante
3: Seja  $t$  um snapshot de  $R$ 
4: Seja  $r$  um resíduo de  $R$ 
5: Seja  $a$  um átomo em  $t$  snapshot
6: Seja  $l$  um átomo de  $L$ 
7: Seja  $MatrizDist$  uma matriz onde cada linha corresponde a um resíduo  $r$  e cada célula
   corresponde à distância entre um  $a$  e  $l$ 
8: Seja  $Result$  uma matriz que armazena, para cada  $t$  snapshot, todas as distâncias mínimas
   entre os  $a$  e  $l$ 
9: Seja  $Input$  uma matriz contendo  $Result$  e, para cada  $t$ , o respectivo valor de FEB.
10: para cada  $t$  em  $TotalSS_R$  faça
11:    $[Result_*] \leftarrow null$ 
12:   para cada  $r$  em  $TotalResiduos_R$  faça
13:      $[MatrizDist_{*,*}] \leftarrow null$ 
14:     para cada  $a$  em  $TotalAtomosConformacao_{R,t}$  faça
15:       para cada  $l$  em  $TotalAtomosLigL$  faça
16:          $Dist_{Ra,Ll} \leftarrow \sqrt{(x_R - x_L)^2 + (y_R - y_L)^2 + (z_R - z_L)^2}$ 
17:          $[MatrizDist_{a,l}] \leftarrow Dist_{Ra,Ll}$ 
18:       fim para
19:     fim para
20:      $[Result_{t,r}] \leftarrow \min([MatrizDist_{r,*}])$ 
21:   fim para
22:    $[Input_{t,*}] \leftarrow [Result_{t,*} + FEB_L]$ 
23: fim para

```

Para ilustrar, a Equação 4.2 apresenta a matriz $[MatrizDist]$ para o resíduo GLY95 do receptor InhA, considerando a preparação do Algoritmo 4.1 para o ligante PIF. Como o resíduo GLY95 possui 7 átomos, tem-se um total de 7 linhas para essa matriz. Já que o ligante PIF tem um total de 24 átomos, essa mesma matriz é composta por um total de 24 colunas (na ilustração são mostradas apenas as 4 primeiras e 4 últimas colunas). É importante mencionar que esta matriz com 168 elementos (7×24) é o resultado da matriz de distância apenas para um resíduo do receptor e um ligante. Nesse sentido, para cada $[MatrizDist]$ recupera-se apenas um único valor para a matriz

[*Result*] (para o exemplo da Equação 4.2, o valor 2,72 destacado). A Tabela 4.4 ilustra a matriz [*Result*] para o complexo InhA-PIF no modelo FFR, a qual contém 30.420 (considerando-se as 10 execuções de docagem para cada conformação, conforme mencionado no capítulo 3) instâncias, correspondentes ao total de conformações utilizadas para o PIF.

$$[MatrizDistancia_{*,*}] = \begin{bmatrix} 7,78 & 7,77 & 5,99 & 5,76 & \dots & 4,22 & 5,83 & 5,73 & 7,77 \\ 8,44 & 8,45 & 6,21 & 5,80 & \dots & 4,65 & 6,50 & 6,44 & 8,06 \\ 6,50 & 6,87 & 5,80 & 5,58 & \dots & 5,50 & 5,62 & 7,02 & 7,70 \\ 7,12 & 7,16 & 6,66 & 6,55 & \dots & 3,81 & 5,95 & 7,72 & 8,44 \\ 5,82 & 5,52 & 4,91 & 4,84 & \dots & \mathbf{2,72} & 4,46 & 4,66 & 6,09 \\ 7,35 & 7,20 & 6,18 & 5,66 & \dots & 3,19 & 6,45 & 6,77 & 7,01 \\ 8,04 & 6,20 & 5,47 & 5,59 & \dots & 6,99 & 7,31 & 7,22 & 7,57 \end{bmatrix} \quad (4.2)$$

Tabela 4.4: Parte da matriz [*Result*] gerada para o ligante PIF

GLY95	...	LYS164	...	THR195	...	LEU268
11,07	...	17,10	...	3,85	...	2,29
11,15	...	17,07	...	5,92	...	4,31
...
2,72	...	3,05	...	5,02	...	2,48
...
9,86	...	5,19	...	4,13	...	2,45

Os arquivos de entrada gerados para cada ligante podem ser vistos como um pré-processamento inicial, mas abrangente, o qual pode ser aprimorado para obedecer às necessidades de cada análise ou experimento de mineração de dados realizado sobre eles.

4.3 Análises sobre os dados armazenados no repositório

Antes de submeter o arquivo de entrada produzido para um algoritmo de mineração, a primeira análise realizada tem por objetivo identificar quais resíduos do receptor que mais interagem para um dado complexo receptor-ligante. O objetivo dessa análise é investigar a importância da flexibilidade explícita do receptor e suas interações intermoleculares com pequenas moléculas. Essa análise concentra-se em identificar resíduos da InhA, considerando o modelo FFR, que mais interagem com os quatro ligantes investigados. Para tanto, a matriz binária [*MatrizBinaria_r*] é gerada a partir da transformação de [*Result*] (Tabela 4.4), num formato binário, indicando se há ou não interação receptor-ligante para para um dado resíduo e a respectiva conformação do receptor. Essa transformação é obtida conforme Equação 4.3, a qual gera um arquivo semelhante ao ilustrado pela tabela 4.5.

$$[MatrizBinaria]_{t,r} = \begin{cases} 0 & \text{se } [Result]_{t,r} > 4 \\ 1 & \text{se } [Result]_{t,r} \leq 4 \end{cases} \quad (4.3)$$

Tabela 4.5: Parte da matriz [Result] gerada para o ligante PIF

GLY95	...	LYS164	...	THR195	...	LEU268
0	...	0	...	1	...	1
0	...	0	...	0	...	0
...
1	...	1	...	0	...	1
...
0	...	0	...	0	...	1

Com a matriz [MatrizBinaria] foi possível somar quantas interações houve para cada resíduo em cada complexo InhA-ligante. De posse desse resultado ordenou-se em ordem decrescente os resíduos que mais interagiram e os 10 primeiros (chamados de *top10*) foram selecionados para cada ligante, conforme ilustrado na Figura 4.3. O objetivo foi de verificar se diferentes ligantes fazem contato em uma mesma região do receptor. A união dos *top10* resíduos para cada ligante está exposto na Tabela 4.6, totalizando 25 resíduos diferentes, sendo que os *top10* resíduos de cada ligante estão destacados. Nessa tabela, cada célula está preenchida com o número de vezes que o resíduo interagiu para cada ligante.

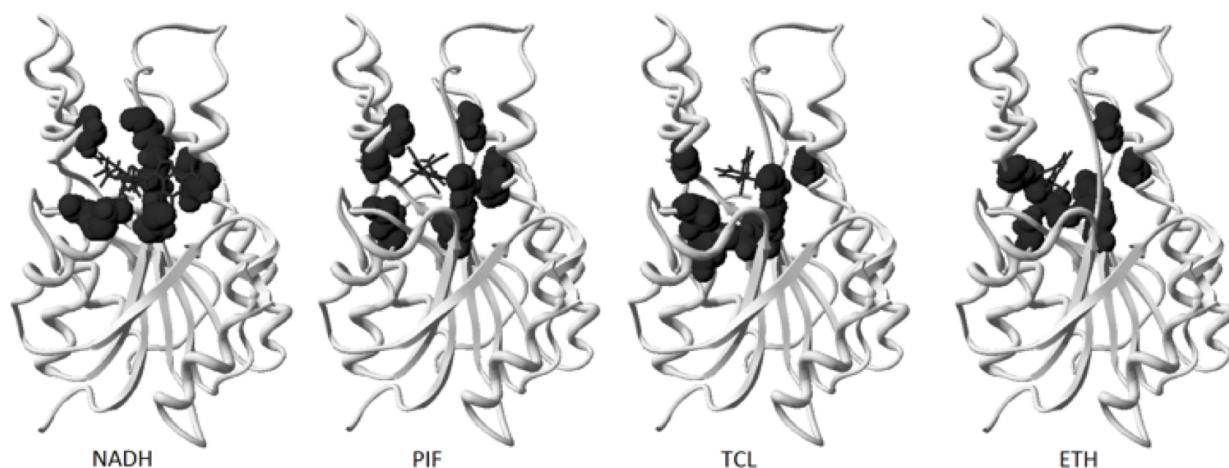


Figura 4.3: Top 10 resíduos do receptor InhA que mais interagem com cada um dos ligantes NADH, PIF, TCL e ETH. O receptor é a estrutura cinza na forma de *Ribbons*. Os 10 resíduos que mais interagem com cada ligante estão na forma de esfera de *van der Walls* e os ligantes na forma de palitos.

Para entender o papel da flexibilidade explícita do receptor, em especial o modelo FFR utilizado neste trabalho, os *top10* resíduos que interagem no modelo FFR foram comparados com os resíduos que interagem quando utilizado apenas a estrutura cristalográfica (PDB:1ENY). Esta seleção foi realizada pela utilização do SPDBV [GUE97]. Tal comparação é apresentada na Tabela 4.7, indicando para cada ligante o número de resíduos que interagiram pelo menos uma vez em todas as execuções

Tabela 4.6: Top 10 resíduos (destacados) para todos os ligantes e suas frequências, totalizando em 25 resíduos. Os *top10* resíduos para cada ligante estão destacados.

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

dos experimentos de docagem (coluna 2); o número de resíduos da estrutura cristalográfica que estão até 4,0 Å de distância do ligante, ou seja, que interagem (coluna 3); e a intersecção dos resíduos que interagem com a estrutura cristalográfica e dos *top10* do modelo FFR.

Tabela 4.7: Comparação do número de resíduos que interagem com cada ligante na estrutura cristalográfica e com os *top10* do modelo FFR.

Ligante	Interações FFR	4,0 Å Estrutura Cristalográfica	Estrutura Cristalográfica \cap Top10
NADH	185	22	9
TCL	139	12	5
PIF	165	13	8
ETH	105	8	4

Pelos dados da Tabela 4.7 é possível identificar a importância em considerar a flexibilidade do receptor em experimentos de docagem molecular. Ao observar, por exemplo, os dados para o ligante NADH, nota-se que o mesmo interage com apenas 22 resíduos da estrutura cristalográfica, enquanto para o modelo FFR o mesmo ligante interage com 185 resíduos. De forma análoga, para o TCL,

139 resíduos interagem com o modelo FFR e apenas 12 com a estrutura cristalográfica e desses apenas 5 também estão presentes no *top10* resíduos. Isso significa que há outros 5 resíduos que podem interagir várias vezes mas que não são identificados se considerar apenas a estrutura rígida do receptor.

4.4 Considerações do capítulo

Considerar a flexibilidade do receptor em experimentos de docagem molecular é um processo que produz uma vasta quantidade de dados que necessitam ser explorados. Para um melhor entendimento desta flexibilidade em experimentos de docagem, neste capítulo foi proposto um repositório suficientemente abrangente que integra conformações de simulação por DM e todos os dados relacionados a respeito das interações receptor-ligante nos seus respectivos resultados dos experimentos de docagem.

Neste capítulo foi mostrado o desenvolvimento do repositório FReDD, o qual foi inicialmente publicado em [WIN09]. O FReDD é capaz de armazenar, indexar e recuperar resultados de docagem molecular. Neste repositório está armazenados dados do receptor InhA e quatro ligantes (NADH, TCL, PIF e ETH). Os testes mostram que a sua implementação contribui para o pré-processamento dos dados, como apresentado em [WIN10c]. Esse pré-processamento possibilitou um análise dos dados, com a qual foi possível extrair informações a respeito da interação ligante receptor, como reportado em [WIN10a]. Por essa análise foi possível identificar relações de interações de resíduos do modelo FFR com os ligantes, sendo que essa análise seria difícil de ser realizada sem uma infraestrutura como o FReDD. O conjunto das características descritas neste capítulo [WIN10b] demonstram o quão efetivo é centralizar esse tipo de dados em um repositório apropriado, de modo com o que o acesso e recuperação dos dados se dá de maneira fácil e clara.

5. EXPERIMENTOS COM MINERAÇÃO DE DADOS

A partir das facilidades em termos de pré-processamento de dados que o repositório FReDD oferece, busca-se aplicar técnicas de mineração de dados sobre esses dados para aumentar o entendimento a respeito do comportamento da flexibilidade do receptor e, assim, contribuir para diminuir a quantidade de execuções de experimentos de docagem molecular. Para tanto, os experimentos de mineração de dados executados durante todo o desenvolvimento desta Tese teve por objetivo responder a seguinte pergunta:

- Como selecionar um subconjunto de conformações que sejam as mais relevantes para indicar se um dado ligante é um composto promissor?

Buscando por diferentes tipos de padrões que pudessem apontar uma direção para responder a essa pergunta, foram aplicadas diferentes técnicas de mineração de dados sobre os dados pré-processados a partir do FReDD, como regras de associação e árvores de decisão para classificação e para regressão.

Os experimentos foram realizados considerando o conjunto de dados inicial, pré-processado pelo Algoritmo 4.1. Entretanto, para cada técnica empregada, esse mesmo conjunto de dados passou por novas etapas de pré-processamento, para que se tornasse apropriado para a tarefa de mineração sendo empregada e seus respectivos objetivos.

Este capítulo apresenta as diferentes técnicas de mineração de dados aplicada sobre os dados armazenados no FReDD, bem como seus respectivos procedimentos de pré-processamento. Além disso, são apresentadas as diferentes avaliações realizadas para esses modelos, e qual conhecimento foi possível extrair a partir dos mesmos. Como resultados dos experimentos realizados e apresentados neste capítulo, obteve-se os seguintes trabalhos científicos:

- Os experimentos realizados com regras de associação estão publicados no *Brazilian Symposium on Bioinformatics* em 2008 [MAC08];
- Os resultados obtidos com árvores de decisão para classificação estão publicados no *Brazilian Symposium on Bioinformatics* de 2010 [MAC10c] e no *IADIS International Conference on Applied Computing* de 2010 [MAC10b];
- Os diferentes resultados obtidos pela realização de experimentos com árvore de decisão para regressão foram publicados no periódico *BMC Genomics* em 2010 [MAC10a] e nas conferências *IADIS International Conference on Applied Computing* de 2010 [WIN10a] e *ISCB Latin America* de 2010 [WIN10c] [MAC10d]. Além disso, o artigo [WIN11] está atualmente submetido ao periódico *International Journal of Data Mining and Bioinformatics* e encontra-se sob revisão;

- Por fim, uma compilação desses trabalhos encontra-se na forma de um capítulo do livro *Tópicos em sistemas colaborativos, multimídia, web e banco de dados* de 2010, o qual foi apresentado como minicurso durante o Simpósio Brasileiro de Banco de Dados em 2010 [WIN10b], bem como na forma de um artigo no periódico *WIRES Data Mining and Knowledge Discovery* em 2011 [MAC11].

5.1 Experimentos com regras de associação

A utilização de regras de associação sobre os dados aqui apresentados tem por objetivo identificar relações de interação entre diferentes resíduos do receptor. Para tanto, utilizou-se o conjunto de dados gerado pelo Algoritmo 4.1, e binarizado conforme Equação 4.3, eliminando-se o atributo alvo (*FEB*). Ou seja, cada célula do conjunto de dados contém o valor 0 quando não há interação com o resíduo (distância $> 4,0\text{\AA}$), e 1 quando há interação. Para cada ligante utilizado nesta Tese foi preparado um arquivo distinto.

Os arquivos preparados foram submetidos ao algoritmo *Apriori* [AGR93], ajustando o valor de suporte para 0,005 e confiança para 0,9, com um número máximo de 1.000 regras. O baixo valor de suporte se justifica pelo alto número de registros com conteúdo distinto no conjunto de dados. Após a geração das regras, as mesmas foram pós-processadas, visando a geração de modelos mais enxutos e eficazes. Algumas regras significativas extraídas estão exemplificadas na Tabela 5.1

Tabela 5.1: Exemplos de regras de associação extraídas dos experimentos

Ligante	Regra
NADH	THR100 = 0 \rightarrow ILE94 = 1
NADH	THR100 = 0 \rightarrow SER19 = 1
NADH	THR100 = 0 \rightarrow THR195 = 1
TCL	ASP93 = 0; GLY95 = 1 \rightarrow SER19 = 1; SER93 = 1
PIF	ARG42 = 0 \rightarrow THR195 = 1
ETH	ILE14 = 0; PHE148 = 1 \rightarrow ALA21 = 0

Nas três primeiras regras para o ligante NADH é possível identificar que para as vezes em que o resíduo THR100 não interage com o NADH, os resíduos ILE94, SER19 e THR195 interagem. Isso significa que, apesar do resíduo THR100 aparentemente não interagir com o receptor, ele se torna representativo para indicar os resíduos que possam interagir.

Muitas outras regras podem ser extraídas. Embora o modelo obtido com regras de associação não estabeleça relação entre os resíduos e valores de FEB, elas podem contribuir para indicar quais os resíduos que mais interagem com o ligante sendo estudado. Isso pode ser útil na busca de novos ligantes para este receptor como, por exemplo, estendendo o trabalho de Quevedo et al. [QUE10].

5.2 Experimentos com árvores de decisão para classificação

Como técnica preditiva, um dos métodos utilizados foi árvore de decisão para classificação. Uma vez em que algoritmos de classificação requerem atributos alvo categóricos, o desafio em aplicar essa técnica para os dados sendo utilizados está na transformação do atributo alvo FEB, o qual é numérico, para valores discretos, distribuídos de maneira adequada ao problema. Para tanto, foram empregadas três técnicas de discretização, as quais foram avaliadas a partir dos modelos de árvore de decisão para classificação induzidos.

5.2.1 Discretização do atributo alvo - FEB

Discretização é o processo de transformar valores contínuos em intervalos de classes que representam esses valores. O procedimento de discretização envolve, basicamente, duas etapas [TAN05]:

1. Decisão do número de categorias. Neste passos os valores do atributo contínuo são ordenados e então divididos em n intervalos, especificados por $n - 1$ pontos de partição;
2. Determinação de como mapear os atributos contínuos para as categorias. Nessa etapa os valores do atributo contínuo são adequadamente mapeados para as classes definidas no passo anterior.

Dentre diferentes métodos de discretização existentes na literatura, utilizaram-se os métodos por igual frequência de intervalos e por igual tamanho de intervalo. Além desses dois métodos, propôs-se a discretização por moda e desvio padrão:

- Método 1: discretização por igual frequência de intervalos. Esse é um método simples que considera que n é o número de intervalos parametrizado e m o número total de instâncias. Assim, os valores contínuos do atributo a ser discretizado são divididos em n intervalos, de modo com que cada intervalo contenha m/n valores, aproximadamente;
- Método 2: discretização por igual tamanho de intervalo. Nessa abordagem, os valores contínuos são divididos em n intervalos parametrizados, onde cada intervalo deve possuir o mesmo tamanho. Para [DOU95], esse é considerado um dos métodos mais simples de discretização, porém vulnerável a pontos discrepantes;
- Método 3. discretização por moda e desvio padrão. Esse método de discretização propõe-se a fazer uma separação dos melhores e piores valores de FEB em classes bem definidas. Para tanto considera-se a moda e o desvio padrão da frequência de distribuição dos valores de FEB sendo discretizados. Para esse método definiu-se um total de 5 intervalos, ou classes, conforme apresentado na equação 5.1 onde x é o atributo a ser discretizado, e M_o e σ representam a Moda e o Desvio Padrão para a distribuição de x . Dessa maneira, ocorrências de melhores e piores casos, as quais são menos frequentes no conjunto de dados, são agrupadas nos intervalos das extremidades da distribuição normal, sendo que ocorrências regulares são distribuídas nos demais intervalos.

$$Classe = \begin{cases} Excelente & \text{se } M_o - 2 * \sigma > FEB \\ Bom & \text{se } M_o - \sigma > FEB \geq M_o - 2 * \sigma \\ Regular & \text{se } M_o + \sigma > FEB \geq M_o - \sigma \\ Ruim & \text{se } M_o + 2 * \sigma > FEB \geq M_o + \sigma \\ MRuim & \text{se } FEB > M_o + 2 * \sigma \end{cases} \quad (5.1)$$

Para os três tipos de discretização utilizados foram parametrizadas 5 classes, sendo elas Excelente, Bom, Regular, Ruim e MRuim (Muito Ruim). A Tabela 5.2 mostra para cada ligante, o número de exemplos (resultados de docagem), o valor médio de FEB e seu respectivo desvio padrão, o valor da Moda e a distribuição dos exemplos em cada classe, para cada um dos 3 métodos.

Tabela 5.2: Distribuição de exemplos nas classes para cada método e cada ligante.

Ligante	Exemplos	FEB	Moda	Método	Excelente	Bom	Regular	Ruim	MRuim
PIF	3.042	$-9,90 \pm 0,60$	-9,90	1	604	607	620	610	601
				2	2.995	26	17	3	1
				3	7	223	2.616	173	23
NADH	2.823	$-12,90 \pm 4,20$	-16,80	1	569	559	565	565	565
				2	757	792	839	408	27
				3	205	1.020	374	903	321
TCL	2.837	$-8,90 \pm 0,30$	-9,00	1	563	556	587	582	549
				2	1.017	1.814	4	0	2
				3	19	158	1.866	645	149
ETH	3.043	$-6,80 \pm 0,30$	-6.70	1	619	591	598	649	586
				2	18	173	1.108	1.531	213
				3	160	512	2.131	226	14

Pela Tabela 5.2 é possível observar que no Método 1, os exemplos estão dispostos nas 5 classes de maneira balanceada. Como o Método 2 distribui os exemplos em um intervalo igual de tamanho, os mesmos podem aparecer desbalanceados. Isso acontece especialmente para os casos dos ligantes PIF e TCL, onde o valor de FEB para este último varia de -10,0 até -4,9 kcal/mol, sendo que o valor de sua Moda é -9,0 kcal/mol, mais próximo do valor mínimo do que do valor máximo de FEB. Além disso, esse mesmo ligante apresenta um desvio padrão de 0,3 kcal/mol, o que significa que o valor de FEB não varia muito, apresentando valores próximos à Moda. Como, para o caso do Método 2, a distribuição para os ligantes PIF e TCL é mais frequente para as classes Excelente e Bom, o modelo de árvore de decisão induzido sobre esse conjunto de dados pode ser distorcido. Por outro lado, ao observar a distribuição do Método 3, apesar de apresentar um grande desbalanceamento nas classes, nota-se que os valores que de fato representam os melhores e piores valores de FEB estão distribuídas nas classes Excelente e MRuim.

5.2.2 Avaliação dos modelos induzidos

Os conjuntos de dados foram submetidos ao algoritmo J48 (implementação do C4.5), parametrizando o número mínimo de instâncias em cada nodo folha para 50, objetivando gerar árvores mais legíveis, requisito importante para o problema e tipo de dados sendo explorados. Os modelos induzidos foram avaliados em termos das métricas típicas utilizadas para árvore de decisão para classificação, como acurácia, tamanho da árvore e Medida-F.

Além disso, introduziu-se uma quarta métrica, a qual indica a taxa de instâncias que pertencem às classes Excelente e Bom. Para essa métrica, aqui denominada TEB, busca-se os menores valores, ou seja, quanto menor a taxa, melhor o resultado.

Os resultados dos modelos estão dispostos na Tabela 5.3, onde cada execução corresponde a uma linha da tabela, e cada coluna mostra o resultado obtido para cada uma das métricas sendo avaliadas. Os melhores valores para cada métrica e cada ligante estão destacados.

Tabela 5.3: Resultados dos modelos de árvore de decisão para classificação.

Método	Ligante	Acurácia	Tam. Árvore	Medida-F	TEB
1	PIF	31,92	71	0,31	39,81
	NADH	61,88	61	0,62	39,96
	TCL	30,49	61	0,30	39,44
	ETH	33,37	77	0,35	39,76
2	PIF	98,68	3	0,98	99,31
	NADH	73,53	43	0,73	54,87
	TCL	64,93	49	0,64	99,79
	ETH	61,02	41	0,57	6,28
3	PIF	86,55	5	0,81	7,56
	NADH	75,41	35	0,75	43,39
	TCL	66,23	17	0,58	6,06
	ETH	70,32	29	0,65	22,08

Pelo Método 1 é possível observar que as métricas foram as piores para todos os ligantes, com exceção da métrica TEB para o ligante NADH. Seus resultados mostram que esse tipo de discretização não é eficiente para dados de docagem molecular.

O Método 2 apresentou os melhores resultados para o ligante PIF. Entretanto, para esse ligante em particular, o método apresenta um total de 99,31% de instâncias nas classes Excelente e Bom (ver Tabela 5.2). Isso significa que o modelo induzido não é útil para extrair informações a respeito dos resíduos envolvidos em bons resultados de docagem, uma vez que quase todas as instâncias do conjunto de dados está classificada como sendo das classes Excelente e Bom. Para o TCL esse mesmo método mostrou um melhor resultado em relação à Medida-F mas, assim como para o PIF, o mesmo método classificou quase todas as instâncias como sendo parte das mesmas classes (Excelente e Bom). Assim, olhando para os resultados de TEB, nota-se distorção nos modelos induzidos.

Por fim, os arquivos discretizados pelo Método 3 obtiveram os melhores resultados dos modelos de árvore de decisão para os ligantes ETH e NADH. Para o ligante TCL este mesmo modelo se mostrou mais efetivo em 3 das 4 métricas utilizadas. Já para o PIF, a métrica que se destacou com esse método foi a TEB.

Pelas árvores de decisão é possível extrair informações a respeito da relação entre os resíduos do Modelo FFR da InhA e as classes de FEB. A Figura 5.1 ilustra o modelo induzido para o complexo InhA-NADH, gerada a partir do conjunto de dados discretizado pelo Método 3.

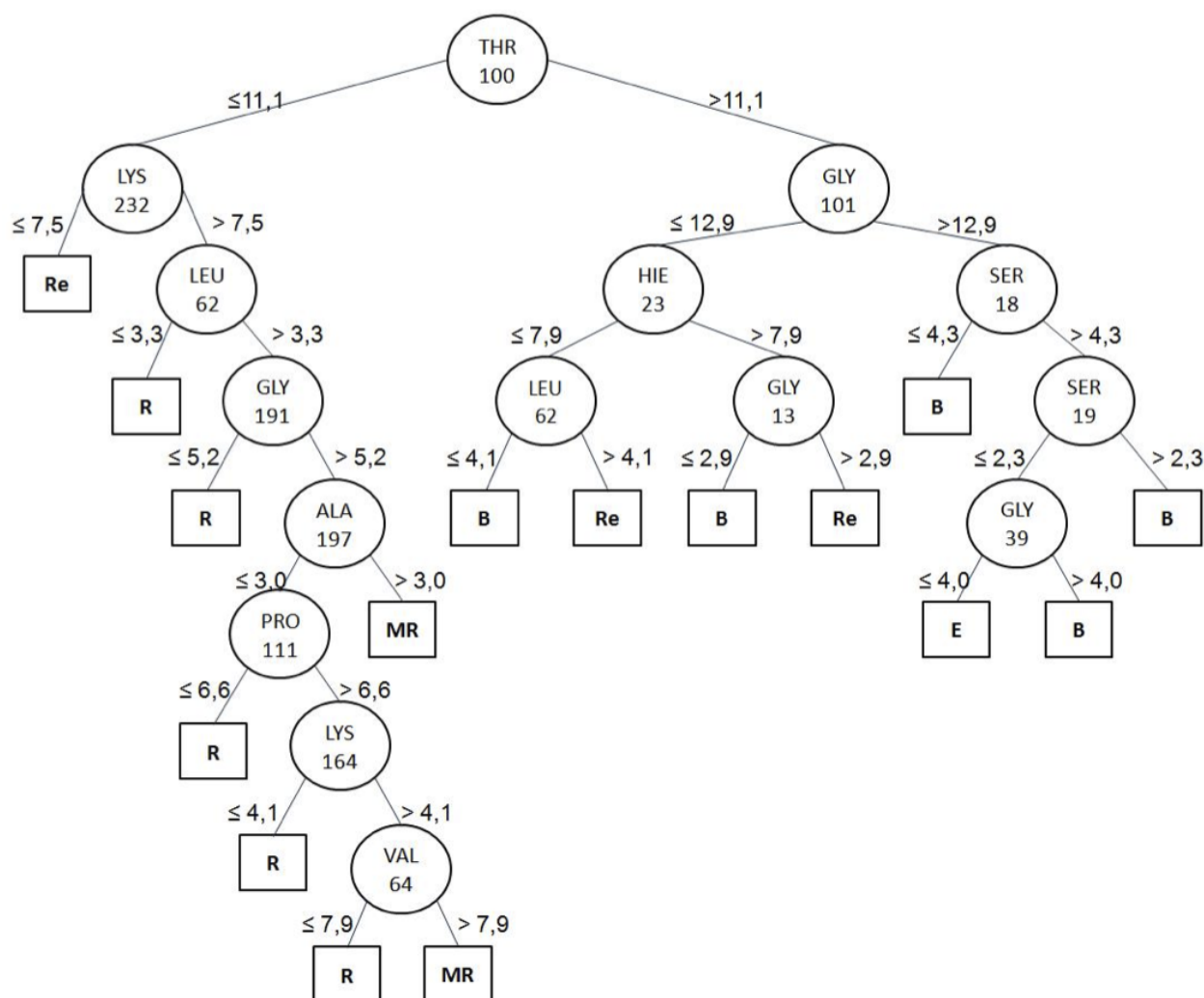


Figura 5.1: Árvore de decisão gerada a partir do arquivo discretizado pelo Método 3 para o complexo InhA-NADH.

Nota-se pela Figura 5.1 que essa árvore está dividindo bem os exemplos, de modo com que todos os exemplos cujas classes sejam Ruim (R) ou Muito Ruim (R) estão à esquerda do nó raiz (resíduo THR100) e todas as instâncias cujas classes estejam associadas a Excelente (E) e Bom (B) estão à direita do nó raiz. Apenas por essa separação já é possível inferir que a posição do resíduo THR100 pode ser fundamental para identificar conformações promissoras para o ligante NADH, isso é, quando o mesmo está a uma distância maior do que 11.0 Å os resultados são promissores. Além

disso, percorrendo a árvore identifica-se que além desse resíduo, as distâncias dos resíduos GLY101, SER18, SER19 e GLY39 podem levar às conformações cuja classe de FEB é Excelente.

5.3 Experimentos com árvores de decisão para regressão

Ao aplicar árvores de decisão para regressão sobre os dados de docagem molecular, busca-se avaliar como diferentes estratégias de pré-processamento podem melhorar a qualidade dos modelos induzidos, bem como melhorar a compreensão dos mesmos. Para tanto, são utilizadas quatro estratégias de processamento:

- Estratégia 1: primeiramente são utilizados o conjunto de dados inicial produzidos conforme o Algoritmo 4.1;
- Estratégia 2: em seguida, busca-se aprimorar esse conjunto de dados, onde é aplicada uma técnica convencional de seleção de atributos;
- Estratégia 3: na busca de um conjunto de dados melhor, é proposta uma outra técnica de seleção desses atributos, a qual é realizada com base no contexto dos dados envolvidos;
- Estratégia 4: por fim, busca-se uma combinação dos atributos selecionados pelas estratégias 2 e 3.

5.3.1 Estratégias de pré-processamento

A Estratégia 1 conta com o conjunto de dados inicial, o qual contém um total de 268 atributos preditivos para cada um dos ligantes sendo testados. Na busca por melhorar a qualidade do conjunto de dados, aplicou-se algumas técnicas de seleção de atributos sobre esses dados.

Primeiramente optou-se, na Estratégia 2, aplicar um algoritmo convencional de seleção de atributos, denominado *Correlation-based Feature Selection* (CFS) [HAL00]. O algoritmo CFS é construído a partir de um método de filtro, o qual ordena as características de acordo com uma função de avaliação baseada em correlação. O objetivo é encontrar um subconjunto de atributos que contenha características fortemente correlacionadas com o atributo-alvo e fracamente correlacionada em relação às demais. O CFS é baseado na seguinte equação:

$$M_X = \frac{K\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}, \quad (5.2)$$

onde M_X é uma função heurística de um subconjunto de dados X que contém k atributos; $\overline{r_{cf}}$ é a média de correlação dos atributos ($f \in X$) e $\overline{r_{ff}}$ é a média entre dois atributos inter-correlacionados.

A Equação 5.2 forma o núcleo do algoritmo CFS. Esse algoritmo foi aplicado para cada um dos conjuntos de dados gerados pelo Algoritmo 4.1 (Estratégia 1), para cada ligante. Como resultado dessa seleção de atributos, obteve-se conjuntos de dados disjuntos, onde o número de atributos preditivos selecionados para cada ligante está listado na Tabela 5.4.

Tabela 5.4: Número de atributos selecionados a partir do algoritmo CFS (Estratégia 2)

Ligante	Atributos Selecionados
NADH	17
TCL	14
PIF	16
ETH	6

Além de uma técnica tradicional de seleção de atributos, buscou-se também aplicar uma técnica com base no contexto dos dados de docagem. A Estratégia 3 baseia-se na definição de Jeffrey [JEF97], o qual diz que a maior distância que permite um contato significativo entre átomos do receptor e do ligante é 4,0 Å. Nesse sentido, partiu-se do princípio de que se um dado resíduo não faz contato em nenhuma das conformações, é pouco provável que este resíduo seja representativo para prever bons valores de FEB. Assim, propôs-se uma estratégia de seleção de atributos considerando tal distância, onde todos os atributos (ou resíduos) cuja distância mínima do conjunto de dados é maior do que 4,0 Å são removidos. O Algoritmo 5.1 ilustra como esse novo conjunto de dados é gerado, a partir do conjunto de dados inicial.

Algoritmo 5.1: Seleção de atributos baseada no contexto de dados de docagem molecular.

```

1: Seja  $R$  um Receptor
2: Seja  $[Input]$  uma matriz que representa o conjunto de dados produzido pelo Algoritmo 4.1
3: Seja  $[InputSA]$  uma matriz contendo o conjunto de dados gerado após a seleção de atributos
4: para cada  $r$  em  $TotalResiduos_R$  faça
5:   se  $\min([Input_{*,r}]) \leq 4$  então
6:      $[InputSA_{*,*}] \leftarrow [[InputSA_{*,*}][Input_{*,r}]]$ 
7:   fim se
8: fim para
9:  $[InputSA_{*,*}] \leftarrow [[InputSA_{*,*}][Input_{*,r+1}]]$ 

```

Aplicando o Algoritmo 5.1 sobre cada conjunto de dados inicial, em vez dos 268 atributos preditivos para cada ligante, obteve-se o número de atributos selecionados ilustrados na Tabela 5.5.

Tabela 5.5: Número de atributos selecionados a partir do Algoritmo 5.1 (Estratégia 3)

Ligante	Atributos Selecionados
NADH	84
TCL	106
PIF	104
ETH	105

Por fim, fez-se a união dos atributos selecionados pelas Estratégias 2 e 3., de modo com que os conjuntos de dados gerados pela Estratégia 4 contêm o número de atributos descritos na Tabela 5.6.

Tabela 5.6: Número de atributos selecionados a partir da combinação das estratégias de seleção de atributos (Estratégia 4)

Ligante	Atributos Selecionados
NADH	93
TCL	114
PIF	108
ETH	111

Os conjuntos de dados foram submetidos ao algoritmo M5P. Dentre os parâmetros disponíveis para este algoritmo, concentrou-se na calibragem dos parâmetros relacionados à legibilidade e precisão das árvores induzidas. Portanto, definiu-se o número mínimo de instâncias para 1.000, onde este tamanho está relacionado com o tamanho da árvore modelo resultante e o número de modelos lineares produzidos.

A Figura 5.2 ilustra a árvore induzida para o complexo InhA-NADH, a qual é composta por 5 nodos e 6 Modelos Lineares (LM). A equação 5.3 ilustra como um modelo linear é composto, onde o valor de FEB é calculado aplicando pesos diferentes para alguns resíduos selecionados do conjunto de dados e calibrando os mesmos com um valor constante. No caso da equação 5.3, é ilustrado o LM6 da árvore da Figura 5.2, por ser o melhor modelo encontrado (mais detalhes na seção seguinte).

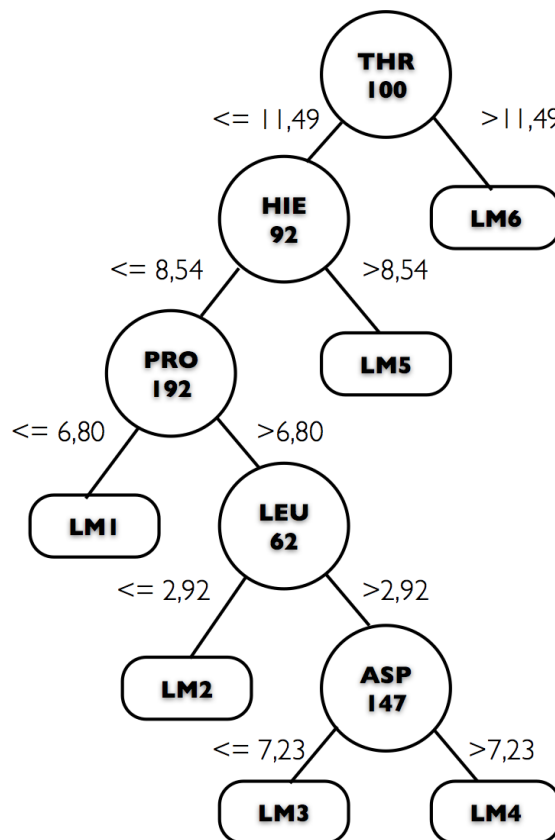


Figura 5.2: Árvore de decisão gerada a partir do arquivo pré-processado pela Estratégia 3, para o complexo InhA-NADH.

$$\begin{aligned}
FEB = & -0,0009 \times SER12 + 0,9405 \times PHE22 + 0,0013 \times THR38 \\
& + 0,0035 \times ASP63 + 0,0006 \times HIE92 + 0,002 \times THR100 \\
& - 0,5005 \times GLY101 - 0,0004 \times ALA123 - 0,0015 \times ASP147 \\
& + 0,0024 \times THR161 + 0,0017 \times LEU167 + 1,094 \times GLY191 \\
& + 0,0037 \times PRO192 + 0,0015 \times ILE193 + 0,0003 \times ILE201 \\
& - 20,6455
\end{aligned} \tag{5.3}$$

5.3.2 Avaliação dos modelos induzidos

Os modelos induzidos foram avaliados considerando-se métricas preditivas e métricas de contexto. Com relação às métricas preditivas, utilizou-se aquelas comuns à avaliação de árvores de regressão, como número de nodos, correlação, MAE e RMSE.

No que diz respeito às métricas de contexto, avaliou-se os modelos considerando os resíduos presentes tanto nos nodos internos quanto os presentes nos modelos lineares de cada modelo, já que o valor estimado de FEB é calculado baseado nos resíduos que fazem parte da grade (Capítulo 3). Uma vez que a mineração de dados sendo aplicada sobre o conjunto de dados de docagem molecular tem como objetivo prever o valor de FEB, é adequado avaliar os modelos que consideram os resíduos que fazem parte desse *grid*.

Um especialista de domínio mapeou todos os resíduos que fazem parte do grid, ou seja, todos os resíduos que pertencem ao sítio ativo de ligação da enzima InhA. Foram selecionados 52 resíduos, aqui denominados *ESR*. Em seguida, mapeou-se, para cada modelo induzido, quais os resíduos que fazem parte dos nodos ou dos modelos lineares (Figura 5.2, Equação 5.3), onde tais resíduos são chamados de *MR*. Para avaliar esses dois conjuntos de resíduos, optou-se por compará-los considerando as métricas de Precisão, Revocação e Medida-F (Equações 2.7, 2.8 e 2.9). No contexto de *ESR* e *MR*, precisão e revocação são assim definidas:

$$Precisão = \frac{ESR \cap MR}{MR} \tag{5.4}$$

$$Revocação = \frac{ESR \cap MR}{ESR} \tag{5.5}$$

A Estratégia 1 foi, talvez, a mais importante em termos de contexto, uma vez que, sem o conhecimento prévio a respeito da semântica dos dados envolvidos, seria difícil gerar um conjunto de dados que pudesse produzir modelos interpretáveis. Ao aplicar as diferentes técnicas de pré-processamento, a idéia é que o pré-processamento baseado em contexto, incluindo a seleção de atributos pela Estratégia 3, pudesse gerar resultados melhores do que aqueles cuja seleção de atributos se desse a partir de técnicas convencionais, como a Estratégia 2.

As Tabelas 5.7 e 5.8 mostram a avaliação das métricas preditivas e de contexto, respectivamente. Essas medidas são individualmente aplicadas para cada ligante. Nas duas tabelas, os melhores valores obtidos estão destacados.

Tabela 5.7: Avaliação do modelo - métricas preditivas

Ligante	Estratégia	Nodos	Correlação	MAE	RMSE
NADH	1	15	0,9536	1,003	1,366
	2	6	0,9483	1,0578	1,4396
	3	5	0,9512	1,0189	1,400
	4	9	0,9513	1,0211	1,3992
PIF	1	22	0,9685	0,3077	0,4071
	2	22	0,9653	0,3237	0,4264
	3	19	0,9692	0,3053	0,4022
	4	19	0,9686	0,3067	0,4060
TCL	1	12	0,9700	0,2396	0,3108
	2	15	0,9667	0,2508	0,3273
	3	19	0,9708	0,2364	0,3068
	4	24	0,9708	0,2369	0,3069
ETH	1	18	0,6086	0,2106	0,2665
	2	16	0,5566	0,2212	0,2790
	3	15	0,5999	0,2123	0,2687
	4	17	0,6047	0,2118	0,2675

Tabela 5.8: Avaliação do modelo - métricas de contexto

Ligante	Estratégia	Precisão	Revocação	Medida-F
NADH	1	0,1176	0,0485	0,0580
	2	0,3636	0,0769	0,1270
	3	0,4375	0,1346	0,2059
	4	0,1875	0,0576	0,0882
PIF	1	0,2143	0,1731	0,1915
	2	0,4667	0,1346	0,2090
	3	0,5294	0,3462	0,4186
	4	0,4571	0,3076	0,3678
TCL	1	0,1282	0,0962	0,1099
	2	0,4286	0,1154	0,1818
	3	0,4412	0,2885	0,3488
	4	0,3928	0,2115	0,2750
ETH	1	0,3939	0,2500	0,3059
	2	0,1250	0,0192	0,0333
	3	0,4375	0,2692	0,3333
	4	0,4516	0,2692	0,3373

Para avaliar os modelos em termos de significância estatística, aplicou-se o Teste de Friedman [SIE88] com um nível de significância $\alpha = 0,05$, aplicado sobre os valores de MAE e RMSE da Tabela 5.7 e sobre o valor de medida-F da Tabela 5.8.

Para as métricas preditivas, foi avaliado se a significância da Estratégia 2 é pior do que as demais. Obteve-se os níveis de significância $p = 0,04$ para MAE e $p = 0,54$ para RMSE, indicando que a Estratégia 2, a qual não utilizou nenhum conhecimento do domínio, é pior do que as demais. Entretanto, esforços ainda precisam ser aplicados sobre esses dados para melhorar sua qualidade.

Por outro lado, no que diz respeito às métricas de contexto, avaliou-se se a Estratégia 3 é significativamente melhor do que as demais. Nesse caso é possível afirmar que sim, pois obteve-se um nível de significância $p = 0,014$, de modo com que é possível inferir que a seleção de atributos baseada no contexto melhora a qualidade dos modelos em relação ao pré-processamento inicial. Essas medidas corroboram com o pressuposto que modelos compreensíveis são essenciais neste contexto.

5.3.3 Pós-processamento dos modelos induzidos

Como o objetivo de minerar os dados de docagem é selecionar conformações promissoras, apenas avaliar a qualidade dos modelos induzidos não é suficiente. Desse modo, estabeleceu-se uma abordagem de pós-processamento das árvores induzidas para seleção de modelos lineares que representem bons valores de FEB. Ao selecionar esses modelos lineares, é possível percorrer a árvore para indentificar as conformações dos conjuntos de dados que pertencem a cada modelo linear. Essa avaliação é realizada em três passos:

- As árvores são percorridas e um teste é aplicado para identificar quais instâncias, ou conformações do conjunto de dados, pertencem à cada nodo folha, ou LM;
- Um critério de seleção de melhores modelos lineares é estabelecido;
- É feita uma avaliação para verificar se as conformações selecionadas são, de fato, promissoras.

Como conjunto de teste, utilizou-se os resultados de docagem com melhores valores de FEB para cada conformação, em vez de fazer uso das 10 execuções (Tabela 4.3, Capítulo 4). Após mapear as conformações que pertencem a cada nodo folha, foi possível estabelecer o critério de seleção de modelos lineares representativos e, assim, utilizá-los para a seleção de conformações:

- Como ponto de partida considerou-se a média dos valores de FEB para cada ligante, para o conjunto de teste (\overline{FEB}_{Teste});
- Em seguida, para cada LM calculou-se a média dos valores de FEB das instâncias que compõe o modelo linear (\overline{FEB}_{LM});
- Tendo esses valores médios, definiu-se que um LM é considerado representativo se a média dos valores de FEB que o compõe é menor ou igual à média dos valores de FEB do conjunto de teste ($\overline{FEB}_{LM} \leq \overline{FEB}_{Teste}$)

Aplicando-se esse critério para a árvore ilustrada na Figura 5.2, foi possível selecionar apenas um modelo linear representativo: LM6 (Equação 5.3). Assim, a partir do modelo gerado para o NADH, pode-se afirmar que o resíduo THR100 é essencial para determinar o valor de FEB para este ligante. Isso é, se o o resíduo THR100 estiver a uma distância maior do que 11,49 Å do NADH, então a conformação provavelmente apresentará um bom valor estimado de FEB, e essa pode ser considerada como uma conformação promissora.

Para os modelos, buscou-se avaliar quais conformações foram selecionadas, bem como verificar se essas selecionadas correspondem, de fato, às melhores. Assim, todas as conformações do conjunto de dados inicial foram ordenadas de acordo com o seu valor de FEB, em ordem crescente, selecionando as primeiras 10 (*top10*), primeiras 100 (*top100*) e primeiras 1.000 (*top1000*). O mesmo foi feito para o conjunto de teste, onde verificou-se quais dessas conformações listadas fazem parte das listadas para o conjunto de dados inicial. Como resultado, obteve-se os dados informados na Tabela 5.9. As colunas 2, 3 e 4 mostram as instâncias do conjunto de teste que pertencem às selecionadas no conjunto de treino, e a coluna 5 indica o total de conformações realmente selecionadas em relação ao total de conformações disponíveis.

Tabela 5.9: Análise dos modelos lineares

Ligante	Top 10	Top 100	Top 1000	Conformações Selecionadas / Conformações
NADH	10	100	998	1.521 / 2.823
TCL	10	100	610	1.780 / 2.737
PIF	10	100	1.000	2.085 / 3.042
ETH	10	92	617	902 / 3.043

Com base nos resultados da Tabela 5.9 é possível notar que a seleção de conformações foi satisfatória para todos os ligantes. Para os ligantes NADH e PIF, dos 10, 100 e 1.000 melhores valores de FEB, o método selecionou quase que 100% das conformações. Apesar de a seleção dos demais ligantes apresentar um valor menor, ela ainda representa aproximadamente 60% do total.

5.4 Considerações sobre os modelos induzidos

Ao analisar os modelos induzidos tanto por regras de associação, quanto por árvore de decisão para classificação e para regressão, considerando-se os resultados para o ligante NADH, nota-se que o resíduo THR100 sempre aparece. Esse é um resíduo que se encontra na alça superior direita da proteína InhA (Figura 3.3, Capítulo 3) e, sendo assim, distante da região do sítio ativo de ligação.

Aparentemente esse é um resíduo que não deveria ser representativo para o entendimento da flexibilidade do receptor e sua relação com os melhores experimentos de docagem. E, de fato, ao observar os modelos induzidos por árvore de decisão (Figuras 5.1 e 5.2), nota-se que o teste das arestas desse resíduo é de aproximadamente 11,00 Å. Essa distância é, de fato, uma distância longa em relação ao sítio ativo e não apresenta nenhum contato. Entretanto, os modelos de árvore de decisão indicam que os melhores resultados de docagem molecular são, justamente, quando esse resíduo está a uma distância superior a 11,00 Å. A partir da análise de um especialista de domínio sobre esses modelos, concluiu-se que esse resíduo é realmente importante para definir conformações que possam resultar em bons resultados de docagem, para o ligante NADH, pois quando o resíduo THR100 está distante do sítio ativo, o mesmo faz com que outros resíduos que formam contato estejam próximos.

Por essa análise, observa-se que os modelos induzidos foram importantes para o entendimento da flexibilidade do receptor e para a identificação das características das conformações, no que diz respeito à distância entre os resíduos do receptor em relação ao ligante, que direcionam à resultados de FEB mais promissores. Contudo, a partir do conjunto de dados sendo utilizado, torna-se difícil selecionar as conformações do receptor para futuros experimentos de docagem. Isso porque as distâncias entre os resíduos do receptor em relação ao ligante só podem ser obtidas a partir de resultados de docagem. Dessa forma, não é possível fazer uso de conformações que não tenham passado por esses experimentos e inferir quais delas teriam mais chance de apresentar bons resultados de FEB após a docagem molecular.

5.5 Considerações do capítulo

Neste capítulo foram apresentadas três técnicas de mineração de dados empregadas nos dados de docagem molecular, onde o principal objetivo foi o de contribuir para a seleção de conformações. Foram utilizadas regras de associação, árvore de decisão para classificação e árvore de decisão para regressão (árvore modelo). Para cada uma dessas técnicas evoluiu-se o pré-processamento inicial apresentado no Capítulo 4. Regras de associação foram aplicadas para identificar quais resíduos interagem mais com o receptor. Essa técnica foi primeiramente utilizada com um conjunto reduzido de dados [MAC08] e foi posteriormente evoluída para utilizar todo o conjunto de dados apresentado [MAC11] [WIN10b]. Ao utilizar árvore de decisão para classificação, propôs-se um método de discretização do FEB [MAC10c] e comparou-se os resultados dos modelos induzidos [MAC10b]. O mesmo foi feito para árvores de decisão para regressão, onde aplicou-se estratégias de pré-processamento sobre esses dados, buscando efetuar uma seleção de atributos baseada no contexto dos dados envolvidos [WIN10c], [WIN11]. Os modelos de árvore de decisão induzidos sobre esses dados [MAC11] foram pós-processados [MAC10a] para identificar a sua qualidade quando da seleção de conformações.

Observou-se que o pré-processamento é uma importante etapa a ser considerada em mineração de dados, onde diferentes técnicas podem ser aplicadas para melhorar a qualidade dos dados minerados. No contexto de dados de docagem molecular observou-se que uma preparação de dados baseada no contexto apresenta-se melhor do que estratégias convencionais de preparação de dados.

Os resultados obtidos com as diferentes técnicas de mineração aplicadas mostram alguns exemplos de informações que podem ser obtidas sobre os experimentos de docagem molecular, que não seria possível de serem extraídas sem a aplicação das técnicas de pré-processamento e rotinas de mineração de dados. Um exemplo são os resíduos que aparecem tanto na árvore de regressão quanto na árvore de decisão do NADH, que são resíduos que em uma inspeção visual com uma conformação desse receptor e o NADH não parecem estar em contato com o mesmo (não estão a uma distância menor do que 4,00 Å do ligante).

Apesar dos bons resultados encontrados, os mesmos não são suficientes para a efetiva seleção das conformações, isso porque não é possível obter as distâncias dos resíduos do receptor em relação ao ligante (requisito do conjunto de dados sendo utilizado) sem ter-se efetuado experimentos de docagem. Nesse sentido, é importante fazer uso de uma estratégia de mineração de dados que permita efetivamente selecionar conformações da proteína de modo que, no futuro, seja possível acelerar os experimentos de docagem molecular, utilizando novos e diferentes ligantes as conformações indicadas como mais promissoras nos experimentos já executados.

6. ALGORITMO 3D-Tri

O processo desenvolvido, incluindo a construção de um repositório alvo, as estratégias de pré-processamento desenvolvidas e os experimentos de mineração de dados, apresentaram resultados interessantes. Esses resultados, entretanto, apesar de promissores ainda podem ser considerados modestos. Nesse sentido, acredita-se ser possível aprimorar os modelos induzidos, mantendo o objetivo de que estes modelos contribuam para a seleção de conformações de receptores para futuros experimentos de docagem molecular.

As estratégias de pré-processamento apresentadas nos capítulos 4 e 5 concentram-se nas distâncias entre átomos do ligante e dos resíduos do receptor. Ainda que essa estratégia tenha sido fundamental para entender e aferir a importância da flexibilidade do receptor, bem como permitir diversos experimentos de mineração de dados sobre esse tipo de dados, seus atributos preditivos demandam uma prévia execução de experimentos de docagem molecular. Em outras palavras, os modelos induzidos indicam quão distante um dado resíduo do receptor precisa estar do ligante sendo testado para que seja atingido um bom valor de *FEB*. Mas, para obter esse valor de distância, é necessário que os experimentos de docagem molecular tenham sido executados. Uma vez em que objetiva-se reduzir o número de conformações do receptor a serem considerados em experimentos de docagem molecular, é interessante que apenas dados de simulação por DM sejam utilizados como atributos preditivos, onde os resultados de docagem molecular sejam considerados apenas no atributo alvo como, por exemplo, fazendo uso dos valores de *FEB*.

Esta Tese apresenta um algoritmo de indução de árvore de decisão para regressão denominado 3D-Tri, o qual é capaz de interpretar propriedades tridimensionais, no formato x, y, z , e induzir uma árvore que representa essas propriedades, predizendo um valor de *FEB*. Para tanto, a estratégia é minerar dados de simulações por DM, considerando as propriedades tridimensionais (3D) de cada conformação do receptor. Isto é, em vez de fazer uso da distância entre os átomos dos resíduos do receptor e os átomos do ligante sendo considerado, assume-se como atributos preditivos as coordenadas espaciais, no espaço euclidiano, de cada átomo dos resíduos do receptor, em cada uma de suas conformações. Em tal estratégia, os valores de *FEB* para cada conformação ainda são considerados como atributo alvo. A proposta desse algoritmo foi submetida e aceita para apresentação no fórum de doutorado da conferência SIAM-SDM (*International Conference on Data Mining*) em 2011.

6.1 Pré-processamento dos dados

A primeira etapa para atender aos objetivos de minerar dados provenientes dos resultados de simulações por DM diz respeito ao pré-processamento desses dados e a geração do conjunto de dados apropriado. Esse conjunto de dados deve conter as conformações tridimensionais dos átomos dos resíduos do receptor para cada conformação. Isto é, para cada receptor identifica-se cada um de

seus átomos e, para cada átomo, obtém-se sua posição espacial x, y, z . O Algoritmo 6.1 apresenta um pseudo código para geração deste conjunto de dados.

Algoritmo 6.1: Geração de um conjunto de dados 3D.

```

1: Seja  $R$  um Receptor
2: Seja  $L$  um Ligante
3: Seja  $t$  uma conformação de  $R$ 
4: Seja  $FEB_{tL}$  o valor de  $FEB$  estimado na conformação  $t$  para o ligante  $L$ 
5: Seja  $TotalSS$  o número de conformações de  $R$ 
6: Seja  $TotalSS_R$  o conjunto de conformações de  $R$ 
7: Seja  $a$  um átomo de  $R$ 
8: Seja  $TotalAtomos$  o número de átomos de  $R$ 
9: Seja  $TotalAtomos_t$  o conjunto de átomos de  $R$ 
10: Seja  $(x, y, z)$  uma coordenada espacial de  $a$ 
11: Seja  $x_{Ra}, y_{Ra}, z_{Ra}$  os valores de  $x, y$  e  $z$  da coordenada espacial do átomo  $a$  de  $R$ 
12: Seja  $Dataset_{t, a \times 3 + 1}$  uma matriz bidimensional de  $t$  linhas e  $a \times 3 + 1$  colunas, contendo cada coordenada espacial de  $a$  na conformação  $t$ , e seu respectivo valor de  $FEB_{tL}$ 
13: para cada  $t$  em  $TotalSS_R$  faça
14:   para cada  $a$  em  $TotalAtomos_t$  faça
15:      $Dataset_{t,*} \leftarrow x_{Ra}$ 
16:      $Dataset_{t,*} \leftarrow y_{Ra}$ 
17:      $Dataset_{t,*} \leftarrow z_{Ra}$ 
18:   fim para
19:    $Dataset_{t,*} \leftarrow FEB_{tL}$ 
20: fim para

```

Cada três colunas de $[Dataset]$ indica uma coordenada espacial de um átomo do receptor. Uma vez que cada linha diz respeito a uma conformação do receptor, o último atributo em $[Dataset]$ é o valor estimado de FEB para a conformação corrente, considerando um dado ligante. Neste sentido, considera-se a execução de um conjunto de dados distinto para cada ligante. Para exemplificar, ao gerar um conjunto de dados para o receptor InhA, considerando todos os seus resíduos, este algoritmo produz um total de 12.024 colunas em $[Dataset]$. A Tabela 6.1 ilustra, para o primeiro e último átomo da InhA, como esse *dataset* seria estruturado para o ligante PIF.

Tabela 6.1: Exemplo de um conjunto de dados gerado para o ligante PIF

$x1$	$y1$	$z1$...	$x4008$	$y4008$	$z4008$	FEB
ALA1_N	ALA1_N	ALA1_N		LEU268_OXT	LEU268_OXT	LEU268_OXT	
15,838	-20,060	8,807	...	-20,647	-17,858	-3,495	-11,22
15,665	-19,974	7,918	...	-20,600	-17,957	-3,176	-11,21
...
14,959	-14,885	-18,370	...	21,498	16,662	-11,545	-1,00

6.2 Algoritmo

O conjunto de dados gerado pelo Algoritmo 6.1 pode não ser corretamente interpretado por um algoritmo convencional de indução de árvore de decisão. Isso é, um algoritmo convencional, ao ler esse conjunto de dados, consideraria cada atributo individualmente, em vez de considerar a relação entre cada três atributos, os quais representam, em conjunto, uma propriedade tridimensional. Em outras palavras, as propriedades tridimensionais do conjunto de dados seriam ignoradas e o modelo induzido poderia não ser preciso, bem como sua interpretação poderia apresentar distorções. Nesse sentido, apresenta-se o Algoritmo 3D-Tri, um algoritmo de indução de árvore capaz de ler o conjunto de dados gerado pelo Algoritmo 6.1, interpretar suas propriedades tridimensionais e induzir uma árvore com base nessas propriedades.

Algoritmos de indução de árvore são basicamente implementados obedecendo a uma estratégia de dividir para conquistar, onde um conjunto de dados X é dividido em regiões locais a fim de prever o atributo alvo. No algoritmo proposto por esta Tese, o principal objetivo é fazer com que essa divisão represente regiões em um espaço euclidiano para um dado atributo preditivo, em função de um valor de FEB .

No contexto do conjunto de dados gerado pelo Algoritmo 6.1, um átomo é um atributo preditivo. Basicamente, para cada nodo da árvore executa-se duas divisões, onde um nodo é um átomo e as arestas testam se os objetos sendo avaliados fazem parte de um dado intervalo $[(x_i, x_f)(y_i, y_f)(z_i, z_f)]$, onde i indica a posição inicial de uma dada coordenada e f representa sua posição final. A Figura 6.1 ilustra esse conceito, sendo que o nodo é um dado átomo e o teste da aresta está em identificar se as instâncias estão *dentro* ou *fora* do intervalo.

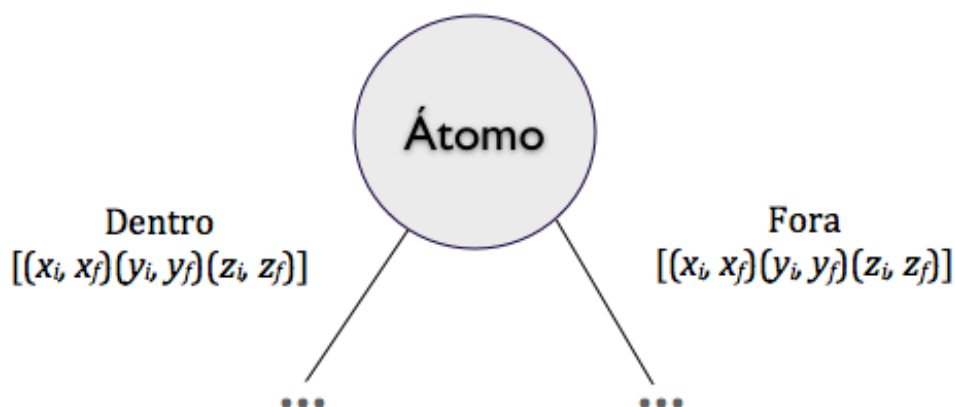


Figura 6.1: Divisão de um nodo pelo pelo algoritmo 3D-Tri

O algoritmo proposto apresenta dois módulos principais:

- O primeiro diz respeito à definição do melhor intervalo, ou bloco, para cada átomo, sendo possível induzir uma árvore binária a partir deste intervalo;
- O segundo refere-se à indução recursiva da árvore a partir das árvores binárias induzidas no módulo anterior.

6.2.1 Definição do bloco

Para gerar as árvores binárias, o primeiro passo é identificar qual o intervalo que um dado átomo deve estar, considerando as instâncias (ou conformações) envolvidas, para que haja um bom valor de FEB . Para tanto, cada átomo de *Dataset* é submetido a uma estratégia de agrupamento, como K-means [HAR79].

O algoritmo K-means assume um valor de k como parâmetro e particiona um dado conjunto de dados X em k grupos, apresentando uma alta similaridade entre objetos em um mesmo grupo, e baixa similaridade entre objetos de grupos distintos. Para tanto, são selecionados k objetos, de maneira aleatória, onde cada um desses objetos representa, inicialmente, uma média ou centro de um dos k grupos. Os objetos restantes são atribuídos ao grupo de maior similaridade, e uma nova média do grupo é calculada, até que haja uma convergência entre os objetos [HAN11]. O critério de convergência é geralmente definido pelo erro quadrático:

$$E = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2 \quad (6.1)$$

onde E é a soma dos quadrados dos erros para todos as instâncias de X , p é o ponto no espaço representando um dado objeto e m_i é a média do grupo C_i .

Para o conjunto de dados *Dataset*, o algoritmo K-means é executado para cada átomo, de modo com que cada um desses átomos seja dividido em k grupos. Para as instâncias de cada k grupo será retornado:

- O valor médio de cada coordenada espacial, obtendo $\bar{x}, \bar{y}, \bar{z}$;
- O valor médio de FEB , definido por \overline{FEB} .

Após a execução do K-means é necessário, para cada átomo, eleger um dos k grupos a ser considerado (GE). Para cada k agrupamento seleciona-se aquele cujas instâncias representam o melhor valor estimado de FEB , ou seja, o menor valor médio de FEB . Dessa maneira define-se o centróide de FEB como:

$$FEB_c = \overline{FEB}_{GE} \quad (6.2)$$

Da mesma forma, o centróide das coordenadas é definido pelo valor médio das coordenadas do agrupamento escolhido:

$$x_c, y_c, z_c = (\bar{x}, \bar{y}, \bar{z})_{GE} \quad (6.3)$$

Em seguida, para cada instância t de *Dataset*, obtem-se a distância Euclidiana (d) entre as coordenadas de t e o centróide das coordenadas (x_c, y_c, z_c):

$$d((x_t, y_t, z_t), (x_c, y_c, z_c)) = \sqrt{(x_c - x_t)^2 + (y_c - y_t)^2 + (z_c - z_t)^2} \quad (6.4)$$

Para fins de ilustração, assume-se que a Tabela 6.2 apresenta um conjunto de dados fictício, onde *Dataset* contém 19 *t* instâncias, composto por um único átomo e seus respectivos valores de *FEB*. Como *t* é um valor dinâmico, cada uma dessas instâncias é representada por um número de conformação *SS*, o qual é invariável. Executando K-means sobre esses dados, onde $k = 2$, os registros de 1 a 9 são agrupados no grupo 1 e os registros de 10 a 19 no grupo 2 (coluna 7). A coluna 8 ilustra o centróide das coordenadas e a coluna 9 mostra o \overline{FEB} para as instâncias de cada grupo. Uma vez que \overline{FEB} do grupo 2 (-7,8) é menor do que \overline{FEB} do grupo 1 (-6,1), o grupo escolhido é o de número 2, de modo com que, para este conjunto de dados, $FEB_c = -7,8$.

Tabela 6.2: Dataset fictício para uma propriedade tridimensional

<i>t</i>	<i>SS</i>	<i>x</i>	<i>y</i>	<i>z</i>	<i>FEB</i>	Grupo	Centróide	\overline{FEB}	<i>d</i>
1	1	1,0	2,0	3,0	-6,0				17,38
2	2	3,0	7,0	9,0	-5,0				10,72
3	3	5,0	8,0	2,0	-7,0				12,85
4	4	7,0	5,0	4,0	-8,0				11,75
5	5	9,0	6,0	7,0	-8,0	1	(5,0, 5,0, 5,0)	$-6,0 \pm 1,8$	8,37
6	6	8,0	3,0	5,0	-7,0				12,08
7	7	2,0	4,0	3,0	-4,0				15,65
8	8	6,0	7,0	4,0	-7,0				11,18
9	9	4,0	3,0	8,0	-3,0				12,69
10	10	14,0	16,0	13,0	-7,0				4,58
11	11	10,0	17,0	14,0	-9,0				5,74
12	12	8,0	10,0	8,0	-11,0				6,00
13	13	11,0	12,0	13,0	-9,0	2	(12,0, 12,0, 12,0)	$-7,8 \pm 2,5$	1,41
14	14	7,0	14,0	12,0	-8,0				5,39
15	15	15,0	9,0	16,0	-4,0				5,83
16	16	13,0	8,0	15,0	-10,0				5,10
17	17	14,0	7,0	12,0	-3,0				5,39
18	18	13,0	13,0	10,0	-9,0				2,45
19	19	15,0	14,0	7,0	-8,0				6,16

A construção do melhor intervalo para cada átomo é feita a partir dos centróides e da distancia Euclidiana calculados. Em um primeiro momento considera-se, para cada coordenada tridimensional, todo o intervalo presente em *Dataset*. Esses intervalos são armazenados em uma estrutura de dados denominada *Bloco*, a qual é composta por $[(x_i, x_f)(y_i, y_f)(z_i, z_f)]$, e definida por

$$\begin{aligned} Coord_i &= \min(Coord_1, \dots, Coord_{TotalSS}) \\ Coord_f &= \max(Coord_1, \dots, Coord_{TotalSS}), \end{aligned} \quad (6.5)$$

onde *TotalSS* é o total de coordenadas, ou instâncias *t*, em *Dataset*. Para esta equação, *Coord* deve ser substituído por cada coordenada *x*, *y* e *z* individualmente.

Essa estrutura de dados será posteriormente atualizada pela expansão dos valores de suas coordenadas. Para a expansão, ordena-se *Dataset* pela distância Euclidiana, em ordem crescente. Caso

exista mais de uma instância t com o mesmo valor de d , a ordenação então se dá pelo menor valor de FEB . Após o ranqueamento é calculado um valor de erro denominado *ErroBloco* para cada instância t , onde esse erro se dá de forma cumulativa por $(FEB_t - FEB_c)$:

$$ErroBloco_t = \sum_{i=1}^t |(FEB_i - FEB_c)| \quad (6.6)$$

A Tabela 6.3 mostra os 12 primeiros registros ordenados por d , (segunda coluna) para as conformações (terceira coluna) da Tabela 6.2. As coordenadas tridimensionais estão dispostas nas colunas 4, 5 e 6 e seu respectivo valor de FEB na coluna 7. Os valores de *ErroBloco* estão na última coluna.

Tabela 6.3: Erro calculado para as instâncias ordenadas pela distância Euclidiana

t	d	SS	x	y	z	FEB	<i>ErroBloco</i>
		Centroid	12,0	12,0	12,0	-7,8	
1	1,41	13	11,0	12,0	13,0	-9,0	1,2
2	2,45	18	13,0	13,0	10,0	-9,0	2,4
3	4,58	10	14,0	16,0	13,0	-7,0	3,2
4	5,10	16	13,0	8,0	15,0	-10,0	5,4
5	5,39	14	7,0	14,0	12,0	-8,0	5,6
6	5,39	17	14,0	7,0	12,0	-3,0	10,4
7	5,74	11	10,0	17,0	14,0	-9,0	11,6
8	5,83	15	15,0	9,0	16,0	-4,0	15,4
9	6,00	12	8,0	10,0	8,0	-11,0	18,6
10	6,16	19	15,0	14,0	7,0	-8,0	18,8
11	8,37	5	9,0	6,0	7,0	-8,0	19,6
12	10,72	2	3,0	7,0	9,0	-5,0	21,8

Para expandir as coordenadas do centróide do grupo escolhido, tendo como base os valores ordenados de d , é necessário estabelecer um critério de parada (*CritParadaBloco*), o qual é baseado em duas métricas. A primeira diz respeito à taxa de crescimento do erro para uma dada instância t , em relação a uma instância anterior (*TaxaErroExpansao*), sendo que esse erro é cumulativo. A segunda refere-se à quantidade de instâncias t , ou população (*TaxaPopExpansao*), que fazem parte da expansão. As taxas de expansão são assim calculadas:

$$TaxaErroExpansao_t = \frac{ErroBloco_t}{ErroBloco_{t-1}} \quad (6.7)$$

$$TaxaPopExpansao_t = \frac{t}{TotalSS} \quad (6.8)$$

O critério de parada baseia-se nessas duas taxas de expansão, comparando-nas com taxas limites de erro (*LimiteErroBloco*) e de população (*LimitePopBloco*) definidas como parâmetro. Assim,

$$CritParadaBloco = \begin{cases} F & \text{se } (TaxaErroExpansao_t \leq LimiteErroBloco) \\ & \text{e } (TaxaPopExpansao_t \leq LimitePopBloco) \\ V & \text{caso contrário} \end{cases} \quad (6.9)$$

Aplicando *CritParadaBloco* no conjunto de dados já ordenados por d , na Tabela 6.3, assumindo que $LimiteErroBloco = 1,5$ e $LimitePopBloco = 0,25$, ignora-se as instâncias $t \geq 6$. Nesse sentido, considera-se as 5 primeiros t instâncias. A Tabela 6.4 ilustra os valores de *TaxaErroBloco* e *TaxaPopBloco* para o conjunto de dados da Tabela 6.3.

Tabela 6.4: Critério de parada para atualização do bloco. Elimina-se a linha 6 e o bloco é atualizado com os valores mínimos e máximos das linhas 1 a 5 para cada coordenada.

t	x	y	z	<i>ErroBloco</i>	<i>TaxaErroExpansao</i>	<i>TaxaPopExpansao</i>
1	11.0	12.0	13.0	1.2	1.0	0.05
2	13.0	13.0	10.0	2.4	2.0	0.10
3	14.0	16.0	13.0	3.2	1.33	0.15
4	13.0	8.0	15.0	5.4	1.68	0.21
5	7.0	14.0	12.0	5.6	1.03	0.26
6	14.0	7.0	12.0	10.4	1.85	0.31

Após aplicar o critério de parada, é possível definir o intervalo $[(x_i, x_f)(y_i, y_f)(z_i, z_f)]$ para um átomo. Assim, para cada coordenada, i é substituído pelo menor valor desta coordenada nas instâncias t após aplicar o critério de parada. Da mesma forma, f é substituído pelo maior valor desta coordenada. Observa-se que, caso o menor ou maior valor de uma determinada coordenada seja igual ao menor ou maior valor desta mesma coordenada no *Dataset* inicial, então assume-se que o limite desta coordenada tende ao infinito. Assim, *Bloco* é atualizado com:

$$Coord_i = \begin{cases} -\infty & \text{se } \min(Coord_1, \dots, Coord_t) = Coord_i \\ \min(Coord_1, \dots, Coord_t) & \text{caso contrário} \end{cases} \quad (6.10)$$

$$Coord_f = \begin{cases} +\infty & \text{se } \max(Coord_1, \dots, Coord_t) = Coord_f \\ \max(Coord_1, \dots, Coord_t) & \text{caso contrário} \end{cases}$$

Onde *Coord* deve ser substituído por x , y , e z individualmente.

Para os dados da Tabela 6.4, $Bloco = [(7, 14)(8, 16)(10, 15)]$. A Figura 6.2 mostra a árvore binária gerada para esse conjunto de dados, sendo que o nodo folha contém o \overline{FEB} para todas as t instâncias de *Dataset* inicial que pertencem a cada uma das arestas da árvore. Além disso, para melhor ilustrar, abaixo dos nodos folha são apresentadas as instâncias do *Dataset* inicial (Tabela 6.2) com suas respectivas coordenadas e valores de *FEB*.

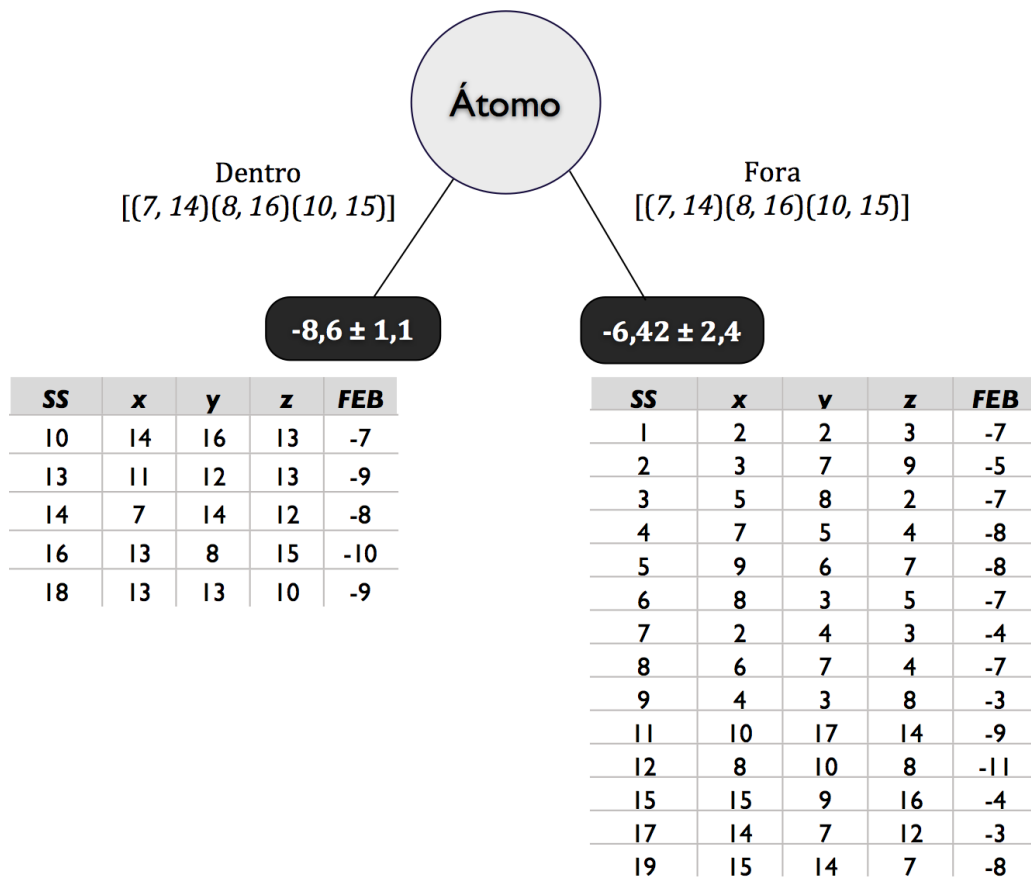


Figura 6.2: Árvore binária induzida

Todo esse processo é repetido para cada átomo de *Dataset*, de modo a existir uma árvore binária, no formato da apresentada na Figura 6.2, para cada um desses átomos. Para a posterior indução da árvore, é necessário escolher um desses átomos. Para tanto, calcula-se o *RDP* (Equação 2.10) de cada bloco, de modo a retornar o bloco com o maior valor de *RDP*.

O Algoritmo 6.2 apresenta o pseudo-código para todo o processo de definição de um bloco para cada átomo de *Dataset*.

6.2.2 Indução da árvore

Tendo-se definido o critério de construção do bloco, é preciso estabelecer o critério para indução da árvore. Para isso faz-se uso da técnica clássica de indução de árvore, a qual é construída recursivamente a partir de uma abordagem *top-down*. O conjunto de dados *Dataset* contém como atributo alvo o valor estimado de *FEB*. Por se tratar de um atributo numérico, trata-se de uma indução de árvore de regressão.

O procedimento principal da indução da árvore recebe como parâmetro o conjunto de dados *Dataset*. Na primeira execução desse procedimento é calculado o valor do desvio padrão do valor de *FEB* de todas as instâncias de *Dataset*

 Algoritmo 6.2: Definição do Bloco.

- 1: Seja $Dataset_{t,a \times 3+1}$ uma matriz bidimensional de t linhas, chamadas instâncias, e $a \times 3 + 1$ colunas, gerada a partir do Algoritmo 6.1
 - 2: Seja DP o desvio padrão calculado para as instâncias sendo computadas
 - 3: Seja a um átomo em $Dataset$, sendo (x, y, z) uma coordenada espacial de a
 - 4: Seja t uma instância em $Dataset$
 - 5: Seja FEB_t o valor de FEB estimado na instância t
 - 6: Seja $Dataset_a$ uma submatriz de $Dataset$ contendo somente as colunas do átomo a e FEB_t
 - 7: Seja $TotalSS$ o número de instâncias em $Dataset$
 - 8: Seja $TotalAtomos$ o número de átomos em $Dataset$
 - 9: Seja RDP a a redução do desvio padrão
 - 10: Seja $Bloco$ uma estrutura de dados contendo $[(x_i, x_f)(y_i, y_f)(z_i, z_f)]$
 - Procedure** DefineBloco($Dataset, DP$)
 - 11: **para** cada a em $TotalAtomos$ **faça**
 - 12: Cria $Dataset_a$ de $Dataset$
 - 13: Divide $Dataset_a$ em k grupos
 - 14: Computa \overline{FEB} para as instâncias de cada grupo
 - 15: Computa $\bar{x}, \bar{y}, \bar{z}$ para cada grupo
 - 16: $GE \leftarrow$ grupo com o menor \overline{FEB}
 - 17: $FEB_c \leftarrow \overline{FEB}_{GE}$
 - 18: $x_c, y_c, z_c \leftarrow (\bar{x}, \bar{y}, \bar{z})_{GE}$
 - 19: **para** cada t em $Dataset_a$ **faça**
 - 20: Computa $d((x_t, y_t, z_t), (x_c, y_c, z_c))$
 - 21: **fim para**
 - 22: **para todos** $TotalSS$ **faça**
 - 23: Inicializa $Bloco$
 - 24: **fim para**
 - 25: Ordena $Dataset_a$ por $ED_1, \dots, ED_{TotalSS} + FEB_1, \dots, FEB_{TotalSS}$ em ordem crescente
 - 26: **repita**
 - 27: Computa $ErroBloco$
 - 28: Computa $TaxaErroExpansao$
 - 29: Computa $TaxaPopExpansao$
 - 30: **até** $CritParadaBloco$
 - 31: Atualiza $Bloco$
 - 32: Computa $RDP(Bloco)$
 - 33: **fim para**
 - 34: **retorna** $(a, Bloco)$ com o maior RDP
-

$$DP = dp(FEB_{Dataset}) \quad (6.11)$$

O valor de DP é utilizado para verificação do critério de parada ($CritParada$) de indução. Este critério avalia o tamanho de $Dataset$ ($Tam_{Dataset}$) em relação a uma população mínima ($PopMinima$) de instâncias que devem fazer parte de uma ramificação, bem como o desvio padrão das instâncias de $Dataset$ sendo avaliado em relação a uma taxa do valor de DP ($TaxaDP$), sendo:

$$CritParada = \begin{cases} V & \text{se } (Tam_{Dataset} < PopMinima) \\ & \text{ou } (dp(Dataset) < TaxaDP * DP) \\ F & \text{caso contrário} \end{cases} \quad (6.12)$$

Obedecendo-se *CritParada*, o processo de indução é então parado para este ramo e é calculado o valor médio de *FEB* de *Dataset* ($\overline{FEB}_{Dataset}$), criando-se um nodo folha para esta aresta da árvore, rotulando tal nodo com $\overline{FEB}_{Dataset}$:

$$Nodo \leftarrow Folha(\overline{FEB}_{Dataset}) \quad (6.13)$$

Se o critério de parada não for obedecido, então o procedimento de definição do bloco é chamado (*DefineBloco*, Algoritmo 6.2), passando como parâmetro *Dataset* e *DP*, retornando para a estrutura de dados *BlocoNodo* o átomo e o seu correspondente intervalo de coordenadas. Assim, um novo nodo é criado na árvore:

$$Nodo \leftarrow BlocoNodo \quad (6.14)$$

Para cada aresta *ar* deste nodo, cujo rótulo é *Bloco*, testa-se as instâncias que estão *dentro* ou *fora* deste intervalo. atualiza-se *Dataset* com essas instâncias para cada aresta ($Dataset_{ar}$), e faz-se uma chamada recursiva do procedimento *InduzArvore*, passando como parâmetro $Dataset_{ar}$.

O Algoritmo 6.3 apresenta o pseudo-código para o processo recursivo de indução da árvore para o conjunto de dados *Dataset*.

A avaliação da árvore induzida pelo Algoritmo 6.3 pode ser realizada seguindo os mesmos critérios tradicionais de indução de árvore de regressão, utilizando as métricas de erro *MAE* (Equação 2.11) e *RMSE* (Equação 2.12).

6.3 Considerações do capítulo

Neste capítulo foi apresentado o algoritmo 3D-Tri, um novo algoritmo de indução de árvore de regressão para propriedades tridimensionais. Este algoritmo foi desenvolvido para ler um conjunto de dados provenientes de resultados de simulação por DM, o qual contém como atributos as coordenadas tridimensionais de átomos de uma determinada proteína, tendo como atributo alvo um valor de *FEB*. Cada instância desse conjunto de dados é uma conformação do modelo flexível dessa proteína. A produção desse conjunto de dados, o qual foi denominado neste capítulo como *Dataset*, está descrita na Seção 6.1 e representada pelo Algoritmo 6.1.

O algoritmo 3D-Tri diferencia-se de abordagens clássicas de indução de árvore, seja para classificação ou para regressão, por dois principais motivos. O primeiro se dá pela mais acurada interpretação do arquivo *Dataset*, o qual possui propriedades dependentes a cada três atributos. Ou seja, cada coordenada no espaço euclidiano (x, y, z) é representada por atributos distintos, e este algoritmo as trata como um único objeto. O segundo ponto pelo qual o algoritmo 3D-Tri se destaca é pela maneira como estes atributos são tratados para indução da árvore. Isso é, estabeleceu-se uma

 Algoritmo 6.3: Indução da Árvore.

- 1: Seja $Dataset_{t,a \times 3+1}$ uma matriz bidimensional de t linhas, chamadas instâncias, e $a \times 3 + 1$ colunas, gerada a partir do Algoritmo 6.1
- 2: Seja $Dataset_n$ uma submatriz de $Dataset$ contendo um subconjunto n de instâncias
- 3: Seja a um átomo em $Dataset$
- 4: Seja DP o desvio padrão calculado para as instâncias sendo computadas
- 5: Seja $PopMinima$ um parâmetro que indica qual a população mínima de instâncias em cada divisão dos nodos
- 6: Seja $TaxaDP$ um parâmetro de erro para a divisão dos nodos
- 7: Seja $BlocoNodo$ uma estrutura de dados que contém $(a, [(x_i, x_f)(y_i, y_f)(z_i, z_f)])$ para a divisão dos nodos

Procedure InduzArvore($Dataset$)

- 8: **se** DP não foi Computado **então**
 - 9: Computa DP
 - 10: **fim se**
 - 11: **se** $CritParada$ **então**
 - 12: Computa $\overline{FEB}_{Dataset}$
 - 13: $Nodo \leftarrow Folha(\overline{FEB}_{Dataset})$
 - 14: **senão**
 - 15: $BlocoNodo \leftarrow DefineBloco(Dataset, SD)$
 - 16: $Nodo \leftarrow BlocoNodo$
 - 17: **para** cada aresta ar de $Nodo$ **faça**
 - 18: $Dataset_{ar} \leftarrow$ instâncias que fazem parte do teste da aresta
 - 19: InduzArvore($Dataset_{ar}$)
 - 20: **fim para**
 - 21: **fim se**
-

estratégia de definição de um intervalo ideal para cada átomo $[(x_i, x_f)(y_i, y_f)(z_i, z_f)]$, de modo com que para cada átomo é induzida uma árvore binária, no formato da apresentada na Figura 6.1, onde o teste das arestas avalia se os atributos pertencem ou não ao bloco definido para este átomo. A leitura de $Dataset$, a definição do bloco e a indução da árvore estão descritas nas Seções 6.2.1 e 6.2.2, e representadas pelos algoritmos 6.2 e 6.3. Os testes para este algoritmo são apresentados no capítulo 7.

7. TESTE DO ALGORITMO 3D-Tri

Neste capítulo é apresentado o teste realizado para o algoritmo 3D-Tri, proposto no Capítulo 6. São detalhados:

- o conjunto de dados utilizado;
- o plano de teste para esse conjunto de dados;
- os resultados do teste realizado.

7.1 Dados utilizados

O teste do algoritmo 3D-Tri foi realizado sobre dados do ligante ETH. O conjunto de dados, denominado *DatasetETH*, foi gerado com base no Algoritmo 6.1. Esse conjunto de dados foi produzido com um total de 12.024 colunas, que correspondem às coordenadas tridimensionais dos 4.008 átomos do receptor, e 3.043 registros, que correspondem às conformações cujas docagens moleculares convergiram para esse ligante (ver Tabela 4.2).

Para fins de redução de dimensionalidade, *DatasetETH* foi pré-processado selecionando-se como atributos apenas os átomos dos top 10 resíduos identificados para este ligante (Tabela 4.6 [WIN10a]), e desconsiderando-se os átomos de hidrogênio (H). Após este pré-processamento, *DatasetETH* passou a ter um total de 229 colunas, que correspondem às coordenadas tridimensionais dos 76 átomos selecionados para os top 10 resíduos. A Tabela 7.1 ilustra as coordenadas x, y, z das três primeiras e três últimas conformações de *DatasetETH*, para seu primeiro e último átomo (Nitrogênio e Oxigênio), os quais fazem parte dos resíduos ILE20 e ILE193, respectivamente. A sigla dos átomos (N e O, no caso da Tabela 7.1) são acompanhadas do número sequencial com que os mesmos aparecem no arquivo PDB da proteína InhA (PDB ID: 1ENY).

Tabela 7.1: Exemplo de coordenadas para *DatasetETH*.

ILE_20	ILE_20	ILE_20	...	ILE_193	ILE_193	ILE_193	FEB
N_134_x	N_134_y	N_134_z	...	O_1439_x	O_1439_y	O_1439_z	
-1,501	-0,553	7,380	...	-10,033	-1,754	6,280	-6,96
-1,654	-0,386	7,494	...	-10,084	-1,541	6,247	-6,76
-1,588	-0,926	7,270	...	-10,590	-1,870	6,428	-6,70
...
-1,722	-0,360	6,726	...	-9,866	-1,801	6,466	-6,34
-1,832	-1,031	7,079	...	-9,847	-2,007	6,384	-6,66
-1,724	-0,650	7,719	...	-10,004	-1,951	6,653	-6,40

Por fim, *DatasetETH* foi dividido em duas partes, sendo uma para Treino (*DatasetETH_Treino*) e outra para teste (*DatasetETH_Teste*). Para o conjunto de teste foram extraídos aproximadamente 3% dos registros de *DatasetETH*, de modo com que *DatasetETH_Treino* contém 2.943 conformações e *DatasetETH_Teste* contém 100 conformações.

7.2 Plano de teste

O plano de teste do algoritmo divide-se em três etapas. A primeira diz respeito à indução do modelo a partir do algoritmo 3D-Tri e os parâmetros utilizados para a indução. A segunda se refere à indução de um modelo de árvore de regressão a partir do algoritmo M5P para comparação dos resultados. A terceira parte corresponde à avaliação dos modelos induzidos e às métricas utilizadas para tal avaliação.

7.2.1 Indução do modelo a partir do algoritmo 3D-Tri

A indução do modelo é feita a partir do conjunto de dados *DatasetETH_Ttreino*. Os parâmetros configurados para execução do teste são:

- **Número de grupos.** Foram definidos dois grupos ($k = 2$) para a identificação do centróide na geração do bloco (Algoritmo 6.2);
- **Taxa de erro na expansão do bloco.** O limite da taxa de erro para o critério de parada da expansão do bloco (Equações 6.7 e 6.9, Algoritmo 6.2) foi definido como 0,5;
- **Taxa de população na expansão do bloco.** O bloco é expandido até atingir o limite de erro do item anterior, ou enquanto o número de exemplos que fazem parte do bloco for inferior a taxa de população mínima. Essa foi definida como uma taxa de 0,05 em relação ao número de exemplos sendo computados (Equações 6.8 e 6.9, Algoritmo 6.2);
- **População mínima para a indução.** Definiu-se um mínimo de 10 exemplos para o critério de parada da indução (Equação 6.12, Algoritmo 6.3);
- **Taxa do desvio padrão para a indução.** Foi definido uma taxa de 0,05 para o desvio padrão dos exemplos sendo computados, para o critério de parada da indução (Equação 6.12, Algoritmo 6.3);
- **Profundidade da árvore.** Para que fosse induzido um modelo enxuto e de fácil interpretação, definiu-se uma profundidade máxima de 5 níveis para a indução da árvore, incluindo os nodos folha.

7.2.2 Indução do modelo a partir do algoritmo M5P

Para a indução do modelo a partir do algoritmo M5P, utilizou-se *DatasetETH*, em vez de *DatasetETH_Treino*. Para que os modelos pudessem ser comparados e serem o mais equivalentes possíveis, os seguintes parâmetros foram configurados:

- **Árvore de regressão.** Optou-se por induzir árvore de regressão em vez de árvores modelo, uma vez que essa última é opção padrão do algoritmo;
- **Número mínimo de instâncias.** Esse parâmetro se refere ao número mínimo de instâncias que devem estar presentes no nodo folha. Esse parâmetro foi calibrado com 600 instâncias para que a árvore apresentasse uma profundidade equivalente à profundidade definida para a indução do modelo pelo algoritmo proposto nesta Tese;
- **Percentual de partição.** Para o teste do modelo habilitou-se a opção de percentual de partição, definindo-se 97%. Com esse valor tem-se o percentual de instâncias para treino e de teste equivalentes aos conjuntos de dados *DatasetETH_Treino* e *DatasetETH_Teste*.

7.2.3 Avaliação dos modelos

Para a avaliação dos modelos induzidos são observadas as seguintes métricas:

- **Erros.** São calculados os erro médio absoluto (MAE, Equação 2.11) e erro médio quadrático (RMSE, Equação 2.12) para as instâncias de teste, aplicando-se o modelo induzido pelas instâncias de treino;
- **Número de nodos.** São observados quantos nodos internos e nodos folha compõe o modelo induzido;
- **Profundidade.** É avaliada qual a profundidade máxima da árvore, considerando-se os nodos folha;
- **Exemplos nos nodos folha.** São verificados o número de exemplos, considerando-se o conjunto de dados *DatasetETH*, que pertencem a cada nodo folha dos modelos induzidos;
- **Distribuição das melhores conformações nos nodos folha.** São ordenadas as 100 melhores conformações (aquelas com valor de FEB mais negativo) para o conjunto de dados *DatasetETH* e avaliado como ocorre a distribuição dessas instâncias nos nodos folha;
- **Semântica.** Além de métricas preditivas, também é avaliada a semântica do modelo induzido, e como o modelo pode ser útil para um especialista de domínio.

7.3 Resultados

As árvores resultantes dos modelos induzidos estão ilustradas nas Figuras 7.1 e 7.2 para o algoritmo 3D-Tri e para o algoritmo M5P, respectivamente.

Para a árvore induzida pelo algoritmo 3D-Tri (Figura 7.1), os nodos indicam o átomo sendo testado, no mesmo formato do cabeçalho do conjunto de dados *DatasetETH*, conforme exemplo da Tabela 7.1. As arestas apontam o teste do intervalo das coordenadas x, y, z (Equação 6.10) identificado para o átomo, e esse intervalo está disposto no centro das duas arestas que dividem o nodo. As arestas à esquerda dos nodos correspondem ao teste das instâncias que pertencem ao intervalo, e as arestas à direita dos nodos correspondem às instâncias que não fazem parte do intervalo. Os nodos folha contém um número de indicação da folha, entre parênteses, e o valor médio de FEB de suas instâncias.

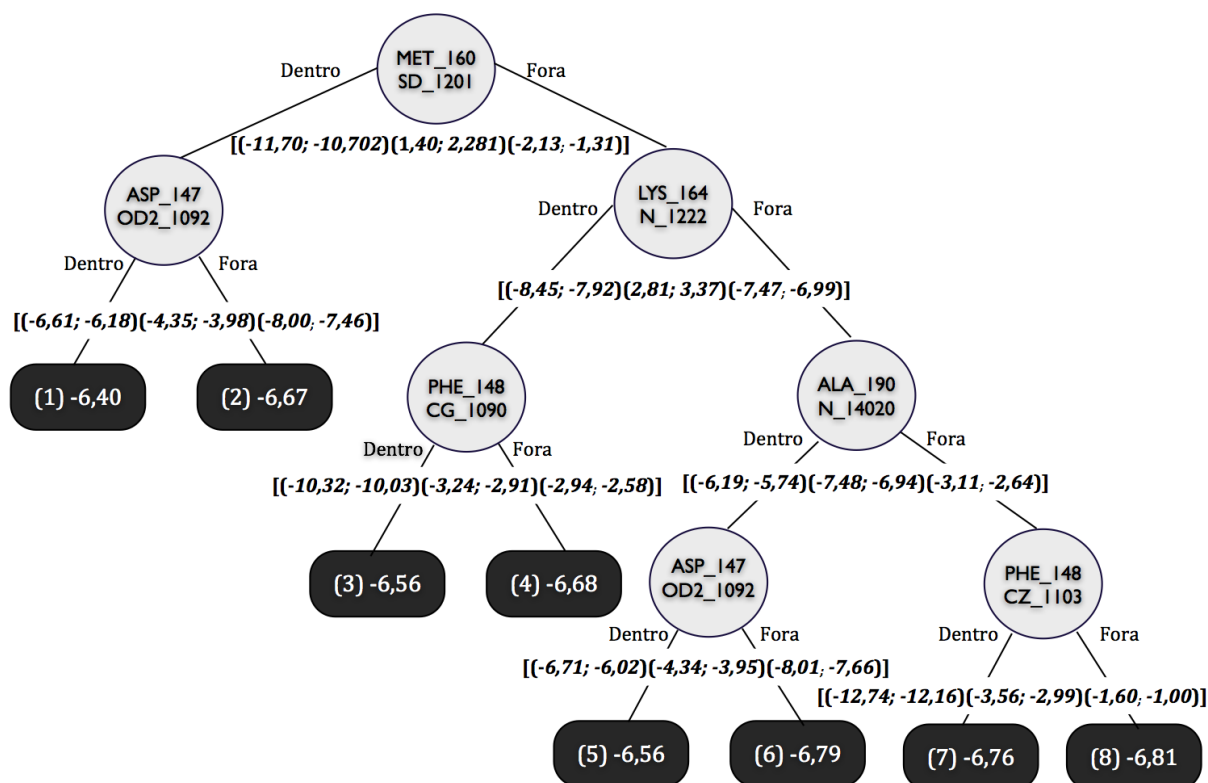


Figura 7.1: Árvore induzida para os top 10 resíduos do ligante ETH pelo algoritmo 3D-Tri

Na árvore induzida pelo algoritmo M5P (Figura 7.2), os nodos representam uma dada coordenada de um átomo do receptor. A descrição do átomo está no mesmo formato do cabeçalho da Tabela 7.1, acompanhado da coordenada sendo testada. As arestas indicam um valor de referência para a coordenada do átomo sendo testado pelo nodo, onde esse valor de referência está disposto no centro das duas arestas que dividem o nodo. O teste das arestas indicam, à esquerda, se a posição da coordenada do átomo é menor ou igual à posição de referência, e à direita se é maior do que o valor de referência. Os nodos folha contém um número de indicação da folha, entre parênteses, e o valor médio de FEB de suas instâncias.

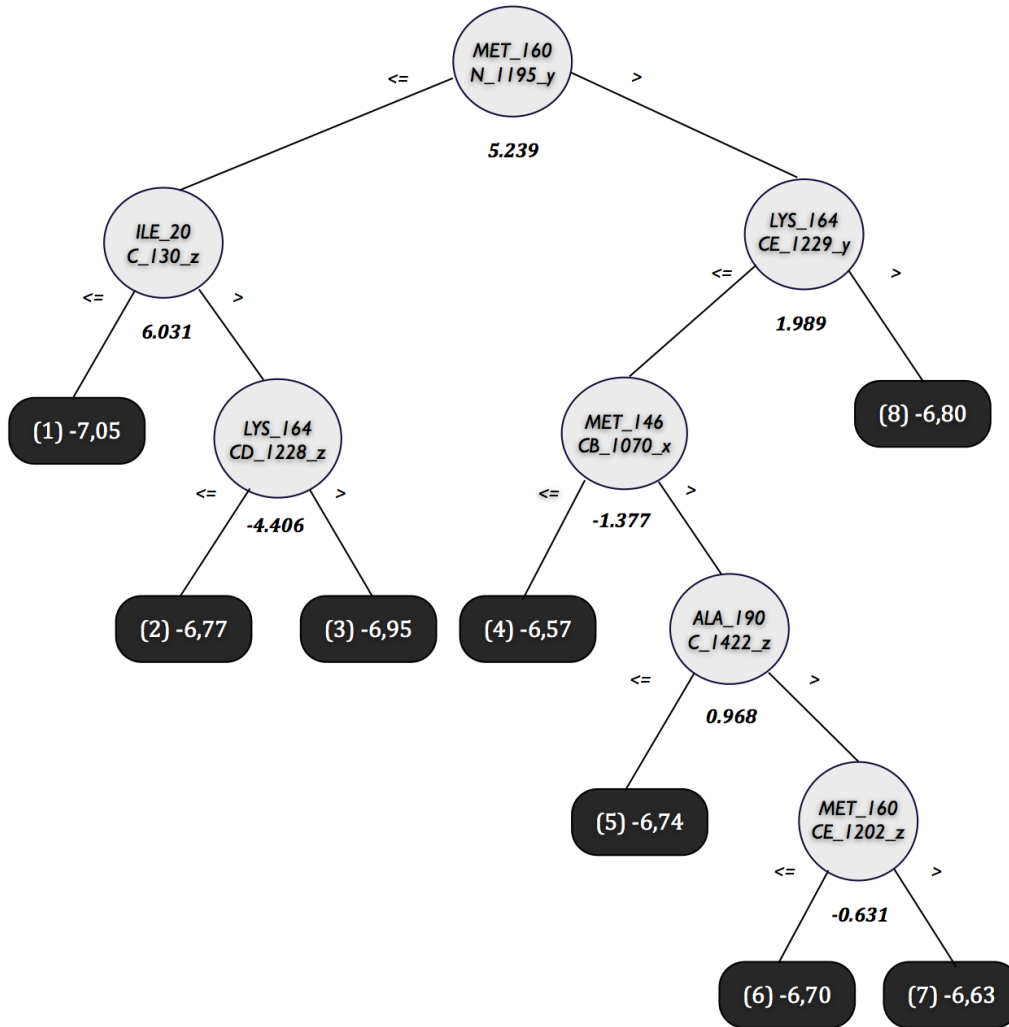


Figura 7.2: Árvore induzida para os top 10 resíduos do ligante ETH pelo algoritmo M5P

As métricas de erro, números de nodo e profundidade das árvores induzidas estão detalhadas na Tabela 7.2, para cada um dos dois algoritmos. A essas métricas dá-se o nome de métricas preditivas. Nota-se, pela Tabela 7.2, que os valores de erro (MAE e RSME) são muito próximos para os dois algoritmos, mas ainda sendo menor para o algoritmo M5P. Além disso, pela calibragem dos parâmetros do M5P, foi possível obter uma árvore equivalente à do algoritmo 3D-Tri em número de nodos e semelhante em relação à profundidade.

Tabela 7.2: Métricas preditivas para os modelos induzidos.

Métrica	3D-Tri	M5P
MAE	0,2513	0,2164
RMSE	0,3184	0,2789
Nodos Internos	7	7
Nodos Folha	8	8
Profundidade	5	6

Apenas a partir das métricas preditivas não é possível inferir qual dos dois modelos apresenta melhor qualidade. Nesse sentido, os modelos são também avaliados em termos do contexto da base de dados *DatasetETH*. Para tanto, avalia-se o número de exemplos em cada nodo folha, e quantos desses exemplos pertencem aos 100 melhores exemplos de *DatasetETH*. Essa avaliação pode ser visualizada na Tabela 7.3.

Tabela 7.3: Métricas de contexto para os modelos induzidos.

Algoritmo	Métricas / Folhas	1	2	3	4	5	6	7	8
3D-Tri	FEB Médio	-6,40	-6,67	-6,56	-6,68	-6,56	-6,79	-6,76	-6,81
	Total Exemplos	11	185	9	176	9	168	157	2328
	Top 100	0	1	0	0	0	3	0	96
M5P	FEB Médio	-7,05	-6,77	-6,95	-6,57	-6,74	-6,70	-6,63	-6,80
	Total Exemplos	402	138	539	339	357	246	514	508
	Top 100	35	3	33	4	1	2	0	22

Pela Tabela 7.3 nota-se que para o algoritmo 3D-Tri, 96 das 100 melhores conformações estão concentradas no nodo folha 8 que é, justamente, o nodo com menor valor médio de FEB. No modelo induzido pelo M5P essas mesmas conformações estão distribuídas entre as folhas da árvore, com maior concentração nas folhas 1, 3 e 8, que também representam os três menores valores médio de FEB para este modelo.

Avaliando-se essas métricas, nota-se que o modelo induzido pelo algoritmo 3D-Tri é promissor, por agrupar as melhores instâncias na mesma folha. Entretanto, o modelo ainda precisa ser expandido para diminuir a concentração de exemplos no nodo folha e verificar a distribuição desses exemplos com essa expansão.

Por fim, é avaliada a semântica dos modelos induzidos. Isso é, é verificado se a árvore pode ser confortavelmente interpretada por um especialista de domínio, e se a mesma pode ser utilizada para a efetiva seleção de conformações para redução do tempo de experimentos em futuros experimentos de docagem molecular.

O modelo induzido pelo algoritmo 3D-Tri tem em seus nodos um átomo e um valor de referência para a posição de suas coordenadas espaciais, até atingir o nodo folha. Por selecionar o nodo folha de menor FEB médio, e sabendo-se que ele concentra as melhores conformações (Tabela 7.3), é possível selecionar conformações a partir das posições dos átomos que fazem parte dos nodos que levam até aos nodos folha escolhidos. Por outro lado, o modelo induzido pelo M5P não trata as coordenadas espaciais dos átomos, indicando um valor de referência apenas para uma das três coordenadas. Apesar de o modelo induzido apresentar bons resultados com relação às métricas preditivas (Tabela 7.2), buscar uma única coordenada de um dado átomo faz pouco sentido para um especialista de domínio, em especial ao tentar identificar esse átomo dentro da estrutura da proteína. Isso se dá, principalmente, porque uma coordenada representa um vetor no espaço, não sendo possível que o especialista de domínio analise o modelo em termos do posicionamento dos átomos e de sua afinidade química. O diferencial do algoritmo 3D-Tri está em permitir ao especialista essa análise tridimensional do átomo no espaço.

7.4 Considerações do capítulo

Este capítulo apresentou um teste para o algoritmo 3D-Tri, comparando seus resultados com os resultados do algoritmo M5P. Foram definidos parâmetros de execução para os dois algoritmos, bem como um plano de avaliação dos modelos. Essa avaliação se deu em termos de métricas preditivas, métricas de contexto e semântica dos modelos induzidos. Pelas métricas preditivas os dois modelos apresentam qualidade semelhante. Pelas métricas de contexto nota-se que o algoritmo 3D-Tri é promissor para seleção de conformações, uma vez que as melhores conformações do conjunto de dados estão agrupadas em um único nodo folha do modelo induzido. No que diz respeito à semântica, entende-se que a árvore induzida pelo algoritmo 3D-Tri pode ser melhor interpretada por um especialista de domínio, de modo com que essa melhor interpretação facilite na seleção de conformações para futuros experimentos de docagem.

8. TRABALHOS RELACIONADOS

Na pesquisa na literatura por trabalhos relacionados, não encontrou-se nenhum que se proponha a minerar dados de docagem molecular para seleção de conformações do receptor, nem trabalhos que utilizam mineração de dados sobre dados tridimensionais como apresentado nesta Tese. Neste capítulo são apresentados os três trabalhos encontrados que apresentam uma maior proximidade em relação ao trabalho desenvolvido nesta Tese. Os trabalhos relacionados são avaliados em termos de:

- **Contexto de RDD.** Verifica-se se o trabalho está inserido em um contexto de RDD;
- **Propriedades Tridimensionais.** É avaliado se o trabalho trata propriedades tridimensionais e como isso é realizado;
- **Tarefa de mineração.** Confere-se se o trabalho utiliza alguma técnica de mineração de dados e, em caso positivo, qual o objetivo de mineração;
- **Utilidade para o problema desta Tese.** Os trabalhos são analisados em termos de sua utilidade e aplicação para o problema desta Tese.

8.1 Banco de dados integrado para RDD

O trabalho desenvolvido por Cockel et al. [COC10], denominado ONDEX, apresenta um repositório de dados para armazenamento e integração de dados para experimentos in silico de descoberta de fármacos. A ideia desse repositório é relacionar dados para a descoberta de novos compostos candidatos. No repositório são integrados dados de diferentes bases disponíveis, como DrugBank, UniProt e BLAST. A relação entre os dados é realizada pela construção de uma rede que contém nodos e arestas, onde os nodos são definidos como conceitos e as arestas são definidas como relações. Os conceitos e relações são capturados das bases de dados integradas, onde essas informações são tratadas de forma textual. Como resultados, é mostrado como a rede de integração pode ser útil para a busca de novos fármacos.

Esse trabalho está inserido no contexto de RDD. Entretanto, não trata as propriedades estruturais das proteínas e ligantes, apenas identifica relações textuais entre elas. Para a construção da rede o trabalho combina diferentes estratégias, como ontologias e técnicas de mineração de textos. No entanto, não é detalhado qual tarefa de mineração foi utilizada. A plataforma ONDEX pode ser utilizada no contexto desta tese para identificar na literatura novos ligantes que tenham chance de serem promissores para experimentos de docagem com a enzima InhA.

8.2 Banco de dados para informações tridimensionais de moléculas

Em Groom e Allen [GRO11] é apresentada uma base de dados para armazenamento de informações tridimensionais de pequenas moléculas, juntamente com informações textuais a respeito de

suas propriedades físico-químicas. O repositório denominado CSD (*Cambridge Structural Database*) foi desenvolvido para facilitar a busca por conhecimento a respeito da interação entre receptores e ligantes, sendo possível identificar a geometria das estruturas e de suas interações intermoleculares.

O ambiente do CSD tem por objetivo armazenar as estruturas tridimensionais das proteínas, buscando contribuir para o entendimento de interações receptor-ligante, mas sem fazer uso de uma abordagem completa de docagem molecular. Ou seja, essa base de dados não relaciona estruturas de proteínas com resultados de experimentos de docagem molecular. Os dados são analisados pelos recursos que a plataforma oferece, sem fazer uso de mineração de dados. Essa plataforma poderia contribuir para o entendimento dos resíduos próximos ao sítio de ligação e, assim, melhorar o pré-processamento dos dados para a execução do algoritmo 3D-Tri.

8.3 Detecção de contatos atômicos em estruturas tridimensionais

A proposta de Toofanny et al. [TOO11] é de identificar contatos entre átomos de uma proteína, através da análise tridimensional das suas conformações, as quais são obtidas por simulações de dinâmica molecular. Nesse sentido, é implementado um índice para acelerar o processo de identificação dessas estruturas na base de dados, onde o objetivo está em reduzir o tempo para descoberta desses contatos. Como resultados é apresentado como esse índice contribuiu para a redução no tempo da identificação dos contatos.

Este trabalho está inserido no contexto de simulações por dinâmica molecular, mas não faz referência a experimentos de docagem sobre as conformações do modelo flexível do receptor. As propriedades tridimensionais das estruturas são essenciais para a construção do índice proposto. Os autores sugerem que pode ser aplicado mineração de dados sobre esses dados futuramente, mas não detalham como isso pode ser feito. O índice proposto pode contribuir para a identificação de novas conformações promissoras, em um modelo de dinâmica molecular mais extenso do que o de 3.100 ps utilizado nessa tese, e tendo por base os dados das distâncias calculadas no pré-processamento dos dados armazenados no FReDD.

8.4 Considerações do Capítulo

Neste capítulo foram apresentados os três trabalhos encontrados na literatura que apresentam uma maior proximidade com o trabalho desta Tese. Por esses trabalhos foi possível identificar que há espaço para pesquisas que consideram as estruturas tridimensionais de proteínas em um contexto de simulação por dinâmica molecular, bem como interesse em realizar pesquisas que fazem uso dessas estruturas para pesquisas em bases e a identificação das relações entre elas. Apesar dos objetivos desses trabalhos relacionados serem diferentes dos objetivos desta tese, a abordagem dos mesmos podem contribuir em algumas das etapas do trabalho desenvolvido nesta Tese.

9. CONCLUSÃO

Esta tese está inserida no contexto de desenho racional de fármacos, onde o principal objetivo é minerar dados de docagem molecular sobre um modelo flexível do receptor, gerado a partir de simulação por dinâmica molecular. Com isso busca-se contribuir para a seleção de conformações promissoras do receptor para um dado tipo de ligante e, assim, reduzir o tempo de execução em novos experimentos de docagem. Os dados utilizados nesta Tese são de um modelo flexível da proteína InhA, do *M. tuberculosis*, considerando quatro ligantes distintos nos experimentos de docagem molecular: NADH, PIF, TCL e ETH.

Durante o desenvolvimento desta Tese foram empregados esforços em fazer uso de diferentes etapas do processo de KDD para tratar os dados envolvidos [WIN10b], onde as principais contribuições estão no desenvolvimento de um repositório alvo para o armazenamento dos dados relacionados aos experimentos de docagem molecular [WIN09] [WIN10a], no pré-processamento desses dados [WIN10c] [WIN11] [MAC10c] e na aplicação de diferentes técnicas de mineração sobre os dados pré-processados [MAC11], como regras de associação [MAC08], árvores de decisão para classificação [MAC10b] e árvores de decisão para regressão [WIN10c] [WIN11] [MAC10a] [MAC10d].

No capítulo 4 foi apresentado o repositório FReDD. Este repositório foi desenvolvido de maneira com que pudesse ser suficientemente abrangente para armazenar, indexar e recuperar resultados de docagem molecular, bem como servir como uma infraestrutura de apoio ao pré-processamento dos dados. Nesse repositório estão armazenados dados a respeito da proteína e dos ligantes sendo considerados nesta Tese. O pré-processamento foi realizado considerando as distâncias mínimas (em Angstroms) entre o ligante e os resíduos do receptor como atributos preditivos, e assumindo o valor de FEB para cada conformação como atributo alvo. Os testes com esse repositório mostram que sua implementação não apenas contribuiu para o pré-processamento dos dados, mas também serviu de apoio para a identificação de padrões a respeito da interação ligante-receptor sobre os dados armazenados. Por essas análises foi possível encontrar relações entre os ligantes utilizados e o modelo flexível do receptor.

O capítulo 5 mostrou como os dados pré-processados a partir do FReDD puderam ser utilizados por diferentes técnicas de mineração de dados. Por regras de associação foi possível extrair regras que estabelecem relações de interações entre os diferentes resíduos do receptor, contribuindo para a identificação, por um especialista de domínio, de quais resíduos do receptor mais interagem com o ligante sendo testado. Por árvores de decisão, seja para classificação ou para regressão, buscou-se extrair modelos que indicassem quais resíduos e sua distância em relação ao ligante contribuem para que o resultado de docagem produza um bom valor de FEB. Ao utilizar árvore de decisão para classificação, propôs-se um método de discretização do FEB e comparou-se os resultados dos modelos induzidos. O mesmo foi feito para árvores de decisão para regressão, onde aplicou-se estratégias de pré-processamento sobre esses dados, buscando efetuar uma seleção de atributos baseada no contexto dos dados envolvidos. Os modelos de árvore de decisão induzidos sobre esses

dados foram pós-processados para identificar a sua qualidade quando da seleção de conformações.

Os resultados obtidos com as diferentes técnicas de mineração aplicadas mostram alguns exemplos de informações que podem ser obtidas sobre os experimentos de docagem molecular, as quais seriam de difícil identificação sem a aplicação das técnicas de pré-processamento e rotinas de mineração de dados. Apesar dos bons resultados encontrados, os mesmos não se mostraram suficientes para a efetiva seleção das conformações. Isso porque não é possível obter as distâncias dos resíduos do receptor em relação ao ligante sem ter-se efetuado experimentos de docagem molecular.

Em direção à uma nova estratégia de mineração para esse contexto, buscou-se considerar como dados de entrada as propriedades tridimensionais de cada átomo do receptor para prever um dado valor de FEB para cada ligante. O capítulo 6 apresentou o algoritmo 3D-Tri, um novo algoritmo de indução de árvore de regressão capaz de identificar as propriedades tridimensionais inerentes ao problema e induzir uma árvore que indique as melhores posições no espaço Euclidiano de determinados átomos que possam resultar em importantes resultados de FEB e, assim, contribuir para a efetiva seleção de conformações do receptor. O algoritmo 3D-Tri diferencia-se de abordagens clássicas de indução de árvore por fazer uso da relação entre as coordenadas x, y, z e considerá-las como um único objeto, bem como pela maneira como esses atributos são tratados para a indução da árvore. Isto é, pela estratégia de definição de um intervalo ideal para cada átomo, representado por $[(x_i, x_f)(y_i, y_f)(z_i, z_f)]$, onde o particionamento do atributo testa se o átomo sendo considerado faz parte ou não deste bloco.

No capítulo 7 foi apresentado o teste do algoritmo 3D-Tri para o ligante ETH, onde utilizou-se as coordenadas dos átomos da proteína para cada conformação obtida por simulação de DM. O modelo induzido pelo algoritmo 3D-Tri foi comparado com um modelo induzido pelo algoritmo M5P. A avaliação considerou métricas preditivas, métricas de contexto e a semântica dos modelos induzidos. As métricas preditivas indicam que os dois modelos têm qualidade semelhantes. As métricas de contexto sugerem que o algoritmo 3D-Tri é promissor para a seleção de conformações por agrupar as melhores conformações do conjunto de dados em um único nodo folha. Por fim, com relação à semântica dos modelos, aquele produzido pelo algoritmo 3D-Tri pode ser melhor interpretado por um especialista de domínio e, assim, facilitar a seleção de conformações para futuros experimentos de docagem molecular. É importante ressaltar que o algoritmo foi parcialmente implementado e, por isso, os testes não foram realizados de maneira exaustiva.

9.1 Publicações

O trabalho desenvolvido durante essa tese atingiu importantes resultados, com os quais foi possível obter as seguintes publicações científicas:

- Dois artigos publicados em periódicos [MAC11] [MAC10a] e um sob revisão [WIN11];
- Nove artigos em conferência, incluindo artigos completos e resumos [WIN10a] [WIN10c] [WIN10d] [MAC10b] [MAC10c] [MAC10d] [QUE10] [WIN09] [MAC08];

- Um capítulo de livro [WIN10b];
- Além disso, a proposta do algoritmo 3D-Tri foi aceita para apresentação no fórum de doutorado do SIAM-SDM 2011.

9.2 Trabalhos futuros

Ao término de uma tese de doutorado, espera-se que o trabalho desenvolvido e resultados obtidos não representem o fim da pesquisa, mas sim um importante avanço para a identificação de novas oportunidades. Com o trabalho apresentado nesta Tese, foi possível identificar diferentes oportunidades relevantes para a continuidade da pesquisa, dentre as quais pode-se citar:

- Implementação completa do algoritmo 3D-Tri;
- Testes mais exaustivos para todos os quatro ligantes, calibrando diferentes valores para os parâmetros definidos no plano de testes;
- Explorar diferentes maneiras de identificação da posição do átomo em relação ao intervalo do bloco;
- Expandir os testes para outros domínios de problema como, por exemplo, para proteínas e ligantes relacionadas a outras doenças de impacto social;
- Examinar diferentes tipos de conformações da região espacial de um átomo, que não apenas um bloco. Dentre essas conformações pode-se investigar como representar esse espaço por uma esfera ou diferentes tipos de superfície.

REFERÊNCIAS

- [AGR93] R. Agrawal, T. Imielinski, A. Swami. "Mining association rules between sets of items in large databases". In: ACM SIGMOD International Conference on Management of Data (SIGMOD93), 1993, pp. 207–216.
- [ALP10] E. Alpaydin. "Introduction to machine learning". Boston: The MIT Press, 2010, 2nd Edition, 584 p.
- [AMA08] R.E. Amaro, R. Baron, J.A. McCammon. "An improved relaxed complex scheme for receptor flexibility in computer-aided drug design". *Journal of Computer-Aided Molecular Design*, vol. 22, 2008, pp. 693–705.
- [BAL09] K.V. Balakin. "Pharmaceutical data mining: approaches and applications for drug discovery". New York: John Wiley & Sons, 2009.
- [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". *Nucleic Acids Research*, vol. 28, 2000, pp. 235–242.
- [BRE84] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone. "Classification and Regression Trees". Belmont: Wadsworth International Group, 1984, 368 p.
- [BRO00] H.B. Broughton. "A method for including protein flexibility in protein-ligand docking: improving tools for database mining and virtual screening". *Journal of Molecular Graphics and Modelling*, vol. 18, 2000, pp. 247–257.
- [CAS07] C.T. Caskey. "The drug development crisis: Efficiency and safety". *Annual Review of Medicine*, vol. 58, 2007, pp. 1–16.
- [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.
- [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: International Conference on Machine Learning, 1995, pp. 194–202.

- [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.
- [FRA11] D. Fraccalvier, A. Pandini, F. Stella, L. Bonati. "Conformational and functional analysis of molecular dynamic trajectories by self-organising maps". *BMC Bioinformatics*, vol. 12, 2011, pp. 1–18.
- [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". *Journal of Molecular Recognition*, vol. 9, 1996, pp. 1–5.
- [GRO11] C.R. Groom, F.A. Allen. "The Cambridge Structural Database: experimental three-dimensional information on small molecules is a vital resource for interdisciplinary research and learning". *WIREs Computational Molecular Science*, vol. 1, 2011, pp. 368–376.
- [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.
- [HAL00] M. Hall. "Correlation-based feature selection for discrete and numeric class machine learning". In: *International Conference on Machine Learning*, 2000, pp. 359–366.
- [HAN11] J. Han, M. Kamber. "Data Mining: Concepts and Techniques". New York: Morgan Kaufmann, 2011, 3rd Edition, 703 p.
- [HAR79] J.A. Hartigan, M.A. Wong. "A K-means clustering algorithm". *Applied Statistics*, vol. 28, 1979, pp. 100–108.
- [IRW05] J.J. Irwin, B.K. Shoichet. "ZINC – a free database of commercially available compounds for virtual screening." *Journal of Chemical Information and Modeling*, vol. 45, 2005, pp. 177–182.
- [JEF97] G.A. Jeffrey. "An introduction to hydrogen bonding". New York: Oxford University Press, 1997, 3rd Edition, 320 p.
- [KAP08] I. Kapetanovic. "Computer-aided drug discovery and development (CADD): in silico-chemico-biological approach". *Chemical-Biological Interaction*, vol. 19, 2008, pp. 1175–1187.

- [KIM09] R. Kimball, M. Ross. "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling". New York: Wiley, 2009, 2nd Edition, 436 p.
- [KUN92] I.D. Kuntz. "Structure-based strategies for drug design and discovery". *Science*, vol. 257, 1992, pp.1078–1082.
- [KUU03] 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". *Journal of Biological Chemistry*, vol. 278, 2003, pp. 20851–20859.
- [LES02] A. Lesk. "Introduction to bioinformatics". New York: Oxford University Press, 2002, 320 p.
- [LIN02] J-H. Lin, A.L. Perryman, J.R. Schames, J.A. McCammon. "Computational drug design accommodating receptor flexibility: The relaxed complex scheme". *Journal of American Chemical Society*, vol. 124, 2002, pp. 5632–5633.
- [LUS01] N.M Luscombe, D. Greenbaum, M. Gerstein. "What is bioinformatics? a proposed definition and overview of the field". *Methods Information in Medicine*, vol. 40, 2001, pp. 346–358.
- [LYB95] T.P. Lybrand. "Ligand-protein docking and rational drug design". *Current Opinion in Structural Biology*, vol. 5, 1995, pp. 224–228.
- [LYN02] P. Lyne. "Structure-based virtual screening: an overview". *Drug Discovery Today*, vol. 7, 2002, pp. 1047–1055.
- [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.
- [MAC08] 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.
- [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.
- [MAC11] K.S. Machado, A.T. Winck, D.D. Ruiz, O. Norberto de Souza. "Mining flexible-receptor docking data". *WIREs Data Mining and Knowledge Discovery*, vol. 1, 2011, pp. 532–541.
- [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". *Journal of Computational Chemistry*, vol. 19, 1998, pp. 1639–1662.
- [MOU04] D.W. Mount. "Sequence and genome analysis". New York: Cold Spring, 2004.
- [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*". *Chemical Communication*, 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.
- [QUE10] C.V. Quevedo, A.T. Winck, K.S. Machado, O. Norberto de Souza, D.D. Ruiz. "A Study of Molecular Descriptor to Rank Candidate Ligands to Inhibit the InhA Receptor". In: International Society for Computational Biology Latin America Conference, 2010, pp. 79–79.
- [QUI86] J.R. Quinlan. "Induction of decision trees". *Machine Learning*, vol. 1, 1986, pp. 81–106.
- [QUI93] J.R. Quinlan. "C4.5: programs for machine learning". San Mateo: Morgan Kaufmann, 1993, 302 p.

- [QUI92] J.R. Quinlan. "Learning with continuous classes". In: Australian Joint Conference on Artificial Intelligence, 1992, pp. 343–348.
- [RAO09] C. B-Rao, J. Subramanian, S.D. Sharma. "Managing protein flexibility in docking and its applications". *Drug Discovery Today*, vol. 14, 2009, pp. 394–398.
- [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". *Biophysical Journal*, vol. 89, 2005, pp. 876–884.
- [SIE88] S. Siegel, N. Castellan Jr.. "Nonparametric statistics for the behavioural sciences". McGraw-Hill, 1988, 2nd Edition, 399p.
- [TAN05] P-N. Tan, M. Steinbach, V. Kumar. "Introduction to data mining". Boston: Addison Wesley, 2005, 2nd Edition, 769.
- [TOO11] R.D. Toofanny, M.S. Simms, D.A.C. Beck, V. Daggett. "Implementation of 3D spatial indexing and compression in a large-scale molecular dynamics simulation database for rapid atomic contact detection". *BMC Bioinformatics*, vol. 12, 2011, pp. 1–10.
- [TOT08] M. Totrov, R. Abagyan. "Flexible ligand docking to multiple receptor conformations: A practical alternative". *Current Opinlon in Structural Biology*, 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.
- [WAN97] Y. Wang, I.H. Witten. "Inducing model trees for continuous classes". In: European Conference on Machine Learning, 1997, pp. 128–137.
- [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". *Journal of Experimental Medicine*, vol. 204, 2007, pp. 73–78.
- [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 inter-molecular 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” . *Tópicos 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.
- [WIN10d] A.T. Winck, K.S. Machado, D.D. Ruiz, V.L. Strube de Lima. “Association Rules to Identify Receptor and Ligand Structures through Named Entities Recognition”. In: International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, 2010, pp. 119–128.
- [WIN11] A.T. Winck, K.S. Machado, O. Norberto de Souza, D.D. Ruiz. “Context-based preprocessing of molecular docking biological data”. *International Journal of Data Mining and Bioinformatics*, Submetido para revisão, 2011, 20p.
- [WIT11] I.H. Witten, E. Frank, M.A. Hall. “Data Mining: Practical Machine Learning Tools and Techniques”. New York: Morgan Kaufmann, 2011 , 3rd Edition, 629 p.