Uma abordagem para a construção de uma única árvore
a partir de uma *Random Forest* para classificação de
bases de expressão gênica

Thais Mayumi Oshiro

DISSERTAÇÃO DE MESTRADO APRESENTADA
À
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM BIOINFORMÁTICA

Programa: Interunidades em Bioinformática

Orientador: Prof. Dr. José Augusto Baranauskas

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro do CNPq

Ribeirão Preto, setembro de 2013

Aos meus pais, Waldemar e Mitsuko,
e meu namorado, Silvio.

Agradecimentos

Agradeço primeiramente a Deus, pelas oportunidades que sempre me forneceu e pelas bênçãos ao longo de minha vida.

Aos meus pais, Waldemar e Mitsuko, a quem devo tudo. Sem eles não chegaria aonde cheguei e não seria metade da pessoa que sou hoje. Agradeço por todo amor e carinho, pela incrível dedicação, pela força e garra, e por todas as oportunidades que sempre me ofereceram.

A toda minha família, por todo o apoio e amor que sempre dedicaram a mim, por sempre acreditar em mim e me incentivar.

Agradeço ao meu namorado, Silvio, por todo o amor, carinho, paciência, amizade e apoio que sempre me deu! Se cheguei até aqui, com toda a certeza, foi porque ele esteve ao meu lado me incentivando sempre!

A meu grande amigo Newton, por toda a sua paciência, companheirismo, alegria, pela sua maravilhosa amizade!

Ao meu orientador, Prof. José Augusto Baranauskas, por ter sido um excelente professor e orientador, por toda a ajuda e apoio no desenvolvimento deste projeto, pela paciência em me ensinar e ajudar e por toda a sabedoria compartilhada.

Resumo

Random Forest é uma técnica computacionalmente eficiente que pode operar rapidamente sobre grandes bases de dados. Ela tem sido usada em muitos projetos de pesquisa recentes e aplicações do mundo real em diversos domínios, entre eles a bioinformática uma vez que a *Random Forest* consegue lidar com bases que apresentam muitos atributos e poucos exemplos. Porém, ela é de difícil compreensão para especialistas humanos de diversas áreas. A pesquisa de mestrado aqui relatada tem como objetivo criar um modelo simbólico, ou seja, uma única árvore a partir da *Random Forest* para a classificação de bases de dados de expressão gênica. Almeja-se assim, aumentar a compreensão por parte dos especialistas humanos sobre o processo que classifica os exemplos no mundo real tentando manter um bom desempenho. Os resultados iniciais obtidos com o algoritmo aqui proposto são promissores, uma vez que ela apresenta, em alguns casos, desempenho melhor do que outro algoritmo amplamente utilizado (J48) e um pouco inferior à *Random Forest*. Além disso, a árvore criada apresenta, no geral, tamanho menor do que a árvore criada pelo algoritmo J48.

Palavras chave: Aprendizado de Máquina, *Random Forest*, Expressão Gênica, Classificador Simbólico.

Abstract

Random Forest is a computationally efficient technique which can operate quickly over large datasets. It has been used in many research projects and recent real-world applications in several fields, including bioinformatics since Random Forest can handle datasets having many attributes, and few examples. However, it is difficult for human experts to understand it. The research reported here aims to create a symbolic model, i.e. a single tree from a Random Forest for the classification of gene expression datasets. Thus, we hope to increase the understanding by human experts on the process that classifies the examples in the real world trying to keep a good performance. Initial results obtained from the proposed algorithm are promising since it presents in some cases performance better than other widely used algorithm (J48) and a slightly lower than a Random Forest. Furthermore, the induced tree presents, in general, a smaller size than the tree built by the algorithm J48.

Keywords: Machine Learning, Random Forest, Gene Expression, Symbolic Classifier.

Lista de Figuras

2.1	Exemplo de curva ROC	10
2.2	Cálculo aproximado da medida AUC	10
2.3	Curva ROC e área AUC do classificador A	11
2.4	Curva ROC e área AUC do classificador B	11
2.5	Validação cruzada utilizando 5 <i>folds</i>	12
2.6	Estrutura de uma árvore de decisão	14
2.7	Funcionamento dos métodos <i>Bagging</i> (a) e <i>Random Forest</i> (b)	17
3.1	Desoxirribose. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)	20
3.2	Nucleotídeo de DNA. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)	21
3.3	Bases nitrogenadas do DNA. Adaptado de (Klug, Cummings, Palladino & Spencer 2010)	21
3.4	Ligação fosfodiéster. Adaptado de (Klug, Cummings, Palladino & Spencer 2010)	21
3.5	Estrutura de DNA. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)	22
3.6	Ribose. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)	23
3.7	Base nitrogenada Uracila. Adaptado de (Klug, Cummings, Palladino & Spencer 2010)	23
3.8	Molécula de tRNA ligada à um códon no mRNA. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)	25
3.9	Expressão gênica. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)	26
3.10	<i>Splicing</i> do RNA. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)	28
3.11	<i>Microarray</i>	32
4.1	Árvore criada pelo classificador J48 utilizando a base DLBCLTumor.	44

4.2	Árvore criada pelo Algoritmo 1 utilizando a base DLBCLTumor.	44
5.1	Resultados das métricas J e JN para as 24 bases.	51
5.2	Resultados das métricas JNP e P para as 24 bases.	51
B.1	Valor AUC em todas as bases de dados (29 bases)	83
B.2	Valor AUC nas 8 bases com baixa densidade	83
B.3	Valor AUC nas 21 bases com alta densidade	83
B.4	Diferenças do valor AUC em todas as bases	83
B.5	Diferenças do valor AUC nas 8 bases com baixa densidade	83
B.6	Diferenças do valor AUC nas 21 bases com alta densidade	83
B.7	Porcentagem de atributos usados em todas as bases	83
B.8	Porcentagem de atributos usados nas 8 bases com baixa densidade	83
B.9	Porcentagem de atributos usados nas 21 bases com alta densidade	84
B.10	Frequência dos 10 atributos mais usados em todas bases. O eixo x corresponde ao número do atributo e o eixo y corresponde à frequência. Embora todos os eixos y tenham ficado em um intervalo de 0 e 1, este intervalo varia em alguns gráficos para melhor visualização.	86
B.11	Frequência dos 10 atributos mais usados em todas bases usando o erro <i>out- of-bag</i> estimado.	87
B.12	Frequência dos 10 atributos mais usados em todas bases usando o erro <i>out- of-bag</i> estimado e <i>minNum</i> = 5.	89

Lista de Tabelas

2.1	Conjunto de exemplos no formato atributo-valor	7
2.2	Exemplo de uma matriz de confusão	9
2.3	Matriz de confusão do classificador A	11
2.4	Matriz de confusão do classificador B	11
3.1	Os 20 aminoácidos com seus respectivos símbolos, abreviaturas e os códons que os codificam	24
5.1	Valores do AUC para os 12 classificadores, desvio padrão e <i>rank</i> médio. . . .	52
5.2	Resultados do teste de Friedman para os valores de AUC para os 12 classifi- cadores.	53
5.3	$\log_{10}$ (Número de nós das árvores para os 12 classificadores) e desvio padrão. .	54
5.4	Resultados do teste de Friedman para os números de nós dos 9 classificadores. .	55
5.5	Valores das métricas de comparação entre árvores geradas pelo Algoritmo 1 (A1-2a(90)) e pelo algoritmo J48	55
5.6	Valores das métricas de comparação entre árvores geradas pelo Algoritmo 1 (A1-2a(90)) e pela <i>Random Tree</i>	56
5.7	Valores dos coeficientes de regressão.	56
A.1	Resumo das bases de dados utilizadas neste experimento, onde n indica o nú- mero de exemplos; c representa o número de classes; a , $a_{\#}$ e a_a indica o número total de atributos, o número de atributos numéricos e o número de atributos nominais, respectivamente; MISS representa a porcentagem de atributos com valores ausentes, sem considerar o atributo classe; as últimas 3 colunas são as métricas de densidade D_1 , D_2 , D_3 de cada base, respectivamente. Bases estão ordenadas por D_2 em ordem crescente.	72
A.2	Resumo das bases de dados utilizadas neste experimento.	75
B.1	Valores AUC, média, mediana e <i>rank</i> médio obtidos nos experimentos. Dados em negrito representam valores excluídos da análise da diferença do AUC. . .	80

B.2	Resultados do teste de Friedman para os valores AUC usando todas bases/8 baixa densidade/21 alta densidade	82
B.3	Resultados do teste de Friedman para os valores AUC usando 128 árvores e considerando um nível de significância de 5%; <i>rank</i> médio para cada valor de <i>minNum</i> e a porcentagem do tempo médio de execução.	88

Sumário

Lista de Figuras	vi
------------------	----

Lista de Tabelas	viii
------------------	------

Sumário	x
---------	---

1	Introdução	1
1.1	Motivação	2
1.2	Objetivo	3
1.3	Organização do Trabalho	4
2	Conceitos Computacionais	5
2.1	Aprendizado de Máquina	5
2.1.1	Conceitos Básicos	6
2.1.2	Árvores de Decisão	11
2.1.3	Métodos <i>Ensemble</i>	13
2.1.4	<i>Bagging</i>	15
2.1.5	<i>Random Trees & Random Forests</i>	16
2.2	Considerações Finais	18
3	Conceitos Biológicos	19
3.1	Biologia Molecular - Conceitos Básicos	19
3.1.1	DNA	19
3.1.2	RNA	22
3.1.3	Expressão gênica	24
3.1.4	<i>Microarrays</i>	30
3.2	Considerações Finais	33

4	Proposta Metodológica	34
4.1	Trabalhos Relacionados	34
4.2	Proposta para construção de uma única árvore.	36
4.3	Métricas de Similaridade de árvores	41
4.4	Considerações Finais	43
5	Resultados & Discussão	46
5.1	Comparação das árvores	49
5.2	Considerações Finais	50
6	Conclusões	57
6.1	Principais resultados e publicações	58
6.2	Trabalhos Futuros	58
	Referências Bibliográficas	59
A	Bases de Dados	71
A.1	Bases de dados usadas para os experimentos realizados	71
A.2	Bases de dados usadas para o Algoritmo 1	73
B	Experimentos Realizados	76
B.1	Métricas de densidade	76
B.2	Experimento 1	78
B.3	Experimento 2	85

Introdução

É sabido que uma combinação de opiniões leva, na maior parte das vezes, a uma decisão melhor do que uma decisão tomada por um único indivíduo. Por exemplo, em um ambiente médico, casos difíceis são melhores solucionados por uma junta composta de vários médicos (e assim, por várias opiniões) do que por um único médico.

Na área de aprendizado de máquina isto também é válido, uma vez que é esperado que a combinação de classificadores apresente um desempenho melhor do que um único classificador (Witten & Frank 1999). Assim, há um grande interesse de pesquisa na área de aprendizado de máquina no que diz respeito a *ensembles* (Dietterich 2000; Kuncheva 2004) — métodos que geram muitos classificadores e combinam os seus resultados. É amplamente aceito que o desempenho de um conjunto de muitos classificadores fracos é geralmente melhor do que um único classificador, dada a mesma quantidade de informação de treinamento (Sirikulviriyaya & Sinthupinyo 2011). Os métodos *ensembles* amplamente conhecidos são *boosting* (Freund & Schapire 1996), *bagging* (Breiman 1996), e mais recentemente *Random Forests* (Breiman 2001; Liaw & Wiener 2002).

Random Forests (Breiman 2001) é um algoritmo *ensemble* proposto por Breiman que

constrói muitas árvores de decisão as quais são utilizadas para classificar um novo exemplo. *Random Forest* apresenta um desempenho excelente em tarefas de classificação e possui características que a tornam ideal para bases de expressão gênica (Díaz-Uriarte & de Andrés 2006).

A análise de dados de expressão gênica é importante para a medicina e biologia uma vez que auxilia no diagnóstico de doenças, ajuda no entendimento da resposta de uma doença à uma droga, proporciona prognósticos precisos para pacientes específicos, entre outros (Klassen, Cummings & Saldaña 2008). Porém, bases de expressão gênica tipicamente possuem muitos atributos e poucas amostras (exemplos) devido à dificuldade de coletar e processar amostras, especialmente para dados obtidos a partir de humanos (Klassen, Cummings & Saldaña 2008). Essa característica dificulta o processo de classificação, pois há muitos atributos irrelevantes e redundantes. Sistemas de aprendizado de máquina são adequados para este problema, mas eles precisam lidar com altos níveis de ruído, uma vez que geralmente, somente um pequeno número de genes é relevante para um determinado problema (Nanni, Brahnam & Lumini 2012).

1.1 Motivação

Atualmente, *Random Forest* é um método de aprendizado *ensemble* amplamente utilizado na literatura e áreas aplicadas. Segundo (Díaz-Uriarte & de Andrés 2006), *Random Forest* apresenta uma série de características que a tornam ideal para bases de dados de expressão gênica: pode ser usada quando há muito mais atributos do que exemplos; pode ser usada para problemas de duas classes ou problemas multi-classe; tem um bom desempenho preditivo mesmo quando a maioria das variáveis preditivas são ruídos e, portanto, não necessita de uma pré-seleção de genes; não superajusta; pode lidar com uma mistura de atributos nominais e numéricos e há pouca necessidade de ajustar os parâmetros para alcançar um bom desempenho.

Porém, mesmo a *Random Forest* apresentando um bom desempenho em altas dimensões, ela é de difícil compreensão para especialistas humanos. Em geral, é mais fácil compreender como uma única árvore de decisão classifica um novo exemplo do que um conjunto de árvores. Neste sentido, o estudo realizado neste trabalho busca um aumento de compreensibilidade

a partir de uma *Random Forest*, tentando manter o bom desempenho da *Random Forest*.

Todavia, deve-se ressaltar que segundo (Gamberger, Lavrac, Zelezny & Tolar 2004), classificadores mais simples, como o aqui proposto, podem apresentar uma qualidade preditiva (desempenho) mais baixa do que classificadores mais complexos, como por exemplo *Random Forest*. Além disso, de acordo com Tan, Steinbach & Kumar (2005) o uso de uma combinação de classificadores aumenta a taxa de acerto se comparado a árvores de decisão obtidas por abordagens mais simples; porém, perde-se uma estrutura facilmente interpretável.

1.2 Objetivo

O objetivo geral do presente trabalho é analisar se a geração de uma única árvore a partir das árvores de uma *Random Forest* consegue manter um desempenho similar ao de uma *Random Forest*, uma vez que é sabido que a *Random Forest* apresenta um bom desempenho em bases de dados de expressão gênica (Pang, Lin, Holford, Enerson, Lu, Lawton, Floyd & Zhao 2006; Díaz-Uriarte & de Andrés 2006). Com isso o trabalho aqui desenvolvido poderá facilitar a compreensão por parte dos especialistas humanos sobre o processo que classifica os exemplos no mundo real.

Os objetivos específicos são:

- analisar se há um número ótimo de árvores em uma *Random Forest*, ou seja, um limiar a partir do qual o aumento do número de árvores não resulta em um ganho significativo do desempenho e somente aumenta o custo computacional;
- analisar a frequência que um atributo aparece no nó raiz de uma *Random Forest* a fim de descobrir se a *Random Forest* utiliza todos os atributos com uma frequência igual ou se há um subconjunto mais utilizado;
- comparar o desempenho da criação de uma árvore, proposta neste trabalho, ao desempenho de outros algoritmos amplamente conhecidos como a *Random Forest*, J48 e *Random Tree*;
- comparar as árvores geradas por diferentes algoritmos a fim de descobrir se suas estruturas (nós utilizados) são semelhantes ou não, por meio de medidas já existentes e medidas que foram propostas neste estudo.

1.3 Organização do Trabalho

O presente trabalho está organizado da seguinte maneira: no Capítulo 2 são apresentados alguns conceitos computacionais utilizados no desenvolvimento deste projeto, incluindo aprendizado de máquina, métodos *ensemble* e *Random Forest*. Os conceitos biológicos utilizados neste projeto, tais como expressão gênica e *microarray*, são apresentados no Capítulo 3. A metodologia aqui proposta e a descrição de experimentos realizados encontram-se no Capítulo 4. No Capítulo 5 são discutidos os resultados obtidos e as conclusões. No Apêndice A são descritas as bases de dados utilizadas no desenvolvimento deste trabalho. No Apêndice B são descritos alguns experimentos adicionais realizados, seus resultados e as conclusões obtidas.

Conceitos Computacionais

Neste capítulo serão apresentados alguns conceitos computacionais utilizados no projeto, como por exemplos conceitos sobre Aprendizado de Máquina, árvores de decisão e *Random Forest*.

2.1 Aprendizado de Máquina

Aprendizado de Máquina (AM) é uma área de Inteligência Artificial (IA) que visa o desenvolvimento de técnicas computacionais capazes de adquirir conhecimento de forma automática. Um sistema de aprendizado é um algoritmo que toma decisões baseado em experiências acumuladas por meio da solução bem sucedida de problemas anteriores (Weiss & Kulikowski 1991).

Por exemplo, para classificar pacientes entre doente ou saudável (com câncer ou não, por exemplo), pode-se analisar o nível de expressão gênica de cada caso, pois as vezes um gene mais ou menos expresso do que o normal pode causar tal doença (câncer).

Aprendizado de Máquina permite obter conclusões genéricas sobre um conjunto particular de exemplos, sendo realizado a partir de raciocínio sobre exemplos fornecidos por um

processo externo ao sistema de aprendizado. O aprendizado de máquina pode ser dividido em supervisionado, não-supervisionado e semissupervisionado (Monard & Baranauskas 2003; Bruce 2001)

O foco deste estudo concentra-se no aprendizado supervisionado, aquele em que classe associada é conhecida (Mitchell 1997). Assim, o algoritmo de aprendizado aprende por meio de um conjunto de entradas e saídas desejadas (classes associadas). Em geral, cada exemplo é descrito por um vetor de atributos e o rótulo da classe a qual este exemplo pertence. Formalmente, no aprendizado supervisionado é fornecida uma característica do objeto a ser alcançada, ou seja, o algoritmo recebe o valor da saída desejada para entrada apresentada.

Os sistemas de aprendizado podem ser divididos ainda em simbólicos e não-simbólicos (Michalski 1983; Kubat, Bratko & Michalski 1998). Os métodos simbólicos ou orientados a conhecimento desenvolvem representações simbólicas do conhecimento, as quais são, geralmente, facilmente interpretadas por seres humanos. São exemplos de métodos simbólicos as árvores de decisão e conjuntos de regras.

Os métodos não-simbólicos ou caixa-preta por sua vez, são caracterizados pelo desenvolvimento de representações próprias do conhecimento, as quais, geralmente, não são facilmente interpretadas por seres humanos. Como exemplos de métodos não-simbólicos, podemos citar as Redes Neurais Artificiais, K-NN e Naive Naves.

2.1.1 Conceitos Básicos

Nesta seção, serão apresentados alguns conceitos básicos de Aprendizado de Máquina utilizados durante o desenvolvimento deste projeto.

Classificador

Dado um conjunto de exemplos de treinamento, um indutor (ou algoritmo de aprendizado) gera como saída um *classificador* (também denominado *hipótese* ou *descrição de conceito*) de forma que, dado um novo exemplo, ele possa prever precisamente sua classe.

Formalmente, em classificação, um exemplo é um par $(x_i, f(x_i))$ onde x_i é a entrada e $f(x_i)$ é a saída. A tarefa de um indutor é, dado um conjunto de exemplos, induzir uma função $h(\cdot)$ que aproxima $f(\cdot)$, normalmente desconhecida. Neste caso, $h(\cdot)$ é chamada uma *hipótese* sobre a função objetivo $f(\cdot)$, ou seja, $h(x_i) \approx f(x_i)$.

Como pode ser observado, no aprendizado supervisionado todo exemplo (x_i, y_i) possui um atributo especial y_i , o rótulo ou *classe*, que descreve o fenômeno de interesse, isto é, a meta que se deseja aprender e poder fazer previsões a respeito. Um exemplo não-rotulado x_i consiste do exemplo, exceto o rótulo, ou seja, um vetor de valores dos atributos. Os rótulos são tipicamente pertencentes a um conjunto discreto (nominal) de classes $\{C_1, C_2, \dots, C_k\}$ no caso de *classificação* ou de valores reais no caso de *regressão*. O foco deste estudo é a classificação.

Um *conjunto de exemplos* é composto por exemplos contendo valores de atributos bem como a classe associada. Na Tabela 2.1 é mostrado o formato padrão de um conjunto de exemplos T com n exemplos e a atributos. Nessa tabela, a linha i refere-se ao i -ésimo exemplo ($i = 1, 2, \dots, n$) e a entrada x_{ij} refere-se ao valor do j -ésimo ($j = 1, 2, \dots, a$) atributo X_j do exemplo i .

	X_1	X_2	$\dots$	X_a	Y
z_1	x_{11}	x_{12}	$\dots$	x_{1a}	y_1
z_2	x_{21}	x_{22}	$\dots$	x_{2a}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
z_n	x_{n1}	x_{n2}	$\dots$	x_{na}	y_n

Tabela 2.1: *Conjunto de exemplos no formato atributo-valor*

Como pode ser notado, exemplos são tuplas $z_i = (x_{i1}, x_{i2}, \dots, x_{ia}, y_i) = (\vec{x}_i, y_i)$ também denotados por (x_i, y_i) , onde fica subentendido o fato que x_i é um vetor. A última coluna, $y_i = f(x_i)$, é a função que tenta-se prever a partir dos atributos. Cada x_i é um elemento do conjunto $X_1 \times X_2 \times \dots \times X_a$ e y_i pertence a uma das k classes, isto é, $y_i \in \{C_1, C_2, \dots, C_k\}$.

Medidas de desempenho

Em Aprendizado de Máquina, utilizam-se algumas medidas para avaliar o desempenho do classificador. Dentre tais medidas, pode-se citar: taxa de erro, taxa de acerto e AUC.

A taxa de erro de um classificador h é denotada por $\text{err}(h)$ e é obtida por meio da Equação 2.1, onde n é o número de exemplos, y_i é a classe verdadeira do exemplo i e $h(x_i)$ é a classe dada pelo classificador h para o exemplo i . Essa medida compara a classe verdadeira de cada exemplo com a classe atribuída pelo classificador h . Se as duas classes forem iguais, ou seja, se $y_i = h(x_i)$, então $\|y_i - h(x_i)\| = 0$; caso contrário, $\|y_i - h(x_i)\| = 1$.

$$err(h) = \frac{1}{n} \sum_{i=1}^n ||y_i \neq h(x_i)|| \quad (2.1)$$

A precisão ou taxa de acerto é denotada por $acc(h)$ e corresponde ao complemento da taxa de erro, conforme é mostrado na Equação 2.2.

$$acc(h) = 1 - err(h) \quad (2.2)$$

As taxas de erro e acerto também podem ser obtidas por meio de uma matriz de confusão. A matriz de confusão é uma matriz cuja dimensão corresponde ao número de classes existentes em um determinado conjunto de exemplos. A sua diagonal principal corresponde ao número de acertos de cada classe e os elementos fora da diagonal principal correspondem ao número de erros. Na Tabela 2.2 é mostrado um exemplo de uma matriz de confusão referente a um conjunto de exemplos com duas classes geralmente denominadas como positiva e negativa.

Nesta tabela, verdadeiros positivos corresponde ao exemplo que é positivo e foi classificado como positivo; falsos positivos são os exemplos negativos que foram classificados como positivos; verdadeiros negativos são os exemplos negativos e que foram classificados como negativos; e falsos negativos são os exemplos positivos que foram classificados como negativos.

A partir da matriz de confusão, pode-se então obter a taxa de erro e a taxa de acerto, representadas pelas Equações 2.3 e 2.4, respectivamente.

$$err(h) = \frac{FN + FP}{VP + FN + FP + VN} \quad (2.3)$$

$$acc(h) = \frac{VP + VN}{VP + FN + FP + VN} \quad (2.4)$$

Outra medida de desempenho, AUC (*area under the ROC curve*), é obtida por meio da geração de um gráfico de sensibilidade *versus* (1-especificidade), conhecido como curva ROC (*receiver operating characteristic*) e calculando a área embaixo da curva (Hand & Till 2001). A sensibilidade da classificação é a razão entre os verdadeiros positivos e o total de

Tabela 2.2: *Exemplo de uma matriz de confusão*

	Predição Positiva	Predição Negativa
Classe Positiva	Verdadeiro Positivo (VP)	Falso Negativo (FN)
Classe Negativa	Falso Positivo (FP)	Verdadeiro Negativo (VN)

exemplos positivos, conforme é mostrado na Equação 2.5. A especificidade da classificação é a razão entre os verdadeiros negativos e o total de exemplos negativos, conforme é mostrado na Equação 2.6.

$$sensibilidade = \frac{VP}{VP + FN} \quad (2.5)$$

$$especificidade = \frac{VN}{FP + VN} \quad (2.6)$$

Na Figura 2.1 é ilustrado um exemplo de curva ROC. Nesta figura, há um único ponto $P(0,4, 0,8)$ e a curva ROC desse ponto é uma curva que tem início no ponto $(0,0)$, passa pelo ponto P e atinge o ponto $(1,1)$. Para calcular um valor AUC aproximado dessa curva, pode-se transformar essa curva em linhas retas, obtendo um gráfico semelhante ao ilustrado na Figura 2.2. Assim, o valor AUC aproximado pode ser calculado por meio da área do trapézio somado à área do triângulo. Dessa forma, o valor AUC aproximado do ponto P seria:

$$AUC = \frac{(1 + 0,8) * 0,6}{2} + \frac{0,4 * 0,8}{2} = 0,54 + 0,16 = 0,7 \quad (2.7)$$

Quanto maior o valor de AUC, melhor é o desempenho do classificador. Os valores de AUC variam em um intervalo $[0, 1]$.

É interessante ressaltar que AUC e precisão nem sempre representam a mesma correlação, ou seja, um alto valor de AUC nem sempre implica em uma alta precisão (Huang & Ling 2005). Por exemplo, considerando um conjunto de exemplos com 10 exemplos positivos e 90 negativos, suponha-se que dois classificadores (A e B) erraram as classes de 8 exemplos, ou seja, a precisão dos dois classificadores será 92%. Entretanto, considerando-se que os 8 erros

Figura 2.1: Exemplo de curva ROC

Figura 2.2: Cálculo aproximado da medida AUC

do classificador A estão entre os exemplos negativos, a taxa de falsos positivos é 9% e a taxa de falsos negativos é 0%. Supondo-se agora que os 8 erros do classificador B estão entre os exemplos positivos, a taxa de falsos positivos será de 0% mas a taxa de falsos negativos será de 80%. Nas Tabelas 2.3 e 2.4, são apresentadas as matrizes de confusão dos classificadores A e B, respectivamente. Com base nesses valores, é possível calcular os valores de sensibilidade e especificidade de cada classificador. Assim, o classificador A possui valor de sensibilidade igual a 1 e especificidade igual a 0,9. Já o classificador B possui sensibilidade igual a 0,2 e especificidade igual a 1. Sendo assim, os dois classificadores possuem diferentes curvas ROC e, consequentemente, diferentes áreas AUC (conforme ilustrado nas Figuras 2.3 e 2.4).

Segundo (Ling, Huang & Zhang 2003) o AUC deveria substituir a precisão (ou taxa de erro, uma vez que esta é igual a $(1 - \text{precisão})$) na comparação de classificadores pois mostrou-se uma medida melhor em extensas comparações experimentais.

Validação Cruzada

Validação cruzada (*r-fold cross validation*) é um método de amostragem utilizado para análise de desempenho que consiste em dividir aleatoriamente os exemplos em r partições mutuamente exclusivas (*folds*) de tamanho aproximadamente igual a n/r exemplos, sendo n o número total de exemplos. Os exemplos nos $r - 1$ *folds* são então usados para treinamento e a hipótese induzida é testada no *fold* restante. Tal processo é repetido r vezes, de modo que todos os *folds* sejam usados uma vez como conjunto de teste, conforme ilustrado na Figura 2.5. Nesta figura, foi utilizado r igual a 5.

Tabela 2.3: *Matriz de confusão do classificador A*

	Predição Positiva	Predição Negativa
Classe Positiva	10 (VP)	0 (FN)
Classe Negativa	8 (FP)	82 (VN)

Tabela 2.4: *Matriz de confusão do classificador B*

	Predição Positiva	Predição Negativa
Classe Positiva	2 (VP)	8 (FN)
Classe Negativa	0 (FP)	90 (VN)

Figura 2.3: *Curva ROC e área AUC do classificador A* **Figura 2.4:** *Curva ROC e área AUC do classificador B*

A seguir, serão apresentados alguns algoritmos e estratégias de AM utilizados no desenvolvimento do presente projeto: Árvore de Decisão, *Bagging*, *Random Forest* e *Random Tree*.

2.1.2 Árvore de Decisão

Árvores de Decisão pertencem a família de algoritmos de AM indutivo *Top Down Induction of Decision Trees*. Uma árvore de decisão utiliza uma estratégia de dividir-para-conquistar, decompondo um problema maior em sub-problemas mais simples, de forma recursiva. A construção de uma árvore de decisão baseia-se na escolha de um atributo que servirá para particionar os exemplos em subconjuntos correspondentes a cada valor do atri-

Figura 2.5: *Validação cruzada utilizando 5 folds*

buto. Passa-se então os exemplos para os subconjuntos de acordo com o valor do atributo escolhido e para cada subconjunto, observa-se se todos os exemplos pertencem a uma mesma classe. Se isso ocorrer, associa-se essa classe à um nó folha contendo todos os exemplos desse subconjunto. Caso contrário, é escolhido um novo atributo para particionar os exemplos e o processo é repetido até que haja uma homogeneidade das classes dos exemplos de um determinado subconjunto. Porém, uma árvore pode crescer muito, tornando-se específica para uma determinada base de dados e causando assim um *overfitting* (super-ajuste).

Poda é uma técnica que evita o problema de *overfitting* por meio da geração de uma hipótese mais genérica a partir do conjunto de treinamento. Como consequência da poda, as folhas não necessariamente conterão exemplos de uma única classe e, assim, a poda pode causar a classificação incorreta de exemplos de treinamento. Há basicamente dois métodos de poda (Baranauskas 2001):

1. pré-poda: durante a geração da hipótese, alguns exemplos de treinamento são delibe-

radamente ignorados, de forma que a hipótese final não classifique todos os exemplos de treinamento corretamente;

2. pós-poda: inicialmente, uma hipótese que explica os exemplos de treinamento é gerada. Após isso, a hipótese é generalizada por meio da eliminação de algumas partes, tais como o corte de alguns ramos em uma árvore de decisão.

Uma árvore de decisão pode ser usada para classificar novos exemplos. Assim, quando desejar-se classificar um novo exemplo, percorre-se a árvore a partir da raiz e desvia-se em cada nó de decisão, segundo o atributo correspondente até que se atinja um nó folha e, então, associa-se a esse novo exemplo a classe correspondente a esse nó folha.

Na Figura 2.6 é ilustrada a estrutura da árvore de decisão do exemplo para classificar o diagnóstico de um novo paciente em saudável ou doente. Começando pela raiz da árvore, o primeiro gene selecionado foi ‘Gene 36’. Se nível de expressão do ‘Gene 36’ ≤ 362 , então segue-se pelo ramo da esquerda da árvore, o que leva a outro gene, ‘Gene 103’. Se o nível de expressão do ‘Gene 103’ ≤ 116 , segue-se pelo ramo da esquerda levando ao ‘Gene 9’. Se o nível de expressão do ‘Gene 9’ ≤ 53 segue-se pelo ramo da esquerda atingindo um nó folha com a classe Saudável associada a ele. Caso o nível de expressão do ‘Gene 9’ > 53 então segue-se pelo ramo da direita chegando no diagnóstico Doente. O processo é repetido até que um nó folha tenha sido alcançado. A classe associada ao nó folha alcançado após percorrer a árvore (saudável ou doente) será a classe (o diagnóstico) associado ao novo paciente.

Árvores de Decisão podem fornecer outras medidas comparativas além das citadas na Seção 2.1.1. Uma delas é o tamanho da árvore, ou seja, o número total de nós (incluindo os nós folhas) que essa árvore possui. Por exemplo, o tamanho da árvore apresentada na Figura 2.6 é 9, pois ela possui quatro nós (correspondendo a quatro genes diferentes) e cinco nós folhas (correspondendo às duas classes do problema). Medidas adicionais são apresentadas na Seção 4.3.

2.1.3 Métodos *Ensemble*

Métodos *ensemble* são algoritmos de aprendizado que constroem um conjunto de classificadores e combinam seus votos para classificar um novo exemplo (Dietterich 2000). O voto corresponde à classe fornecida por cada um dos classificadores para esse novo exem-

Figura 2.6: Estrutura de uma árvore de decisão

plo. Os métodos *ensembles* amplamente conhecidos são *boosting* (Freund & Schapire 1996), *bagging* (Breiman 1996) e *Random Forests* (Breiman 2001; Liaw & Wiener 2002).

O método *boosting* cria diferentes classificadores base reponderando sequencialmente os exemplos no conjunto de treinamento. No início, todos os exemplos são inicializados com pesos iguais. Cada exemplo classificado erroneamente pelo classificador anterior recebe um peso maior na próxima iteração do treinamento, a fim de tentar classificá-lo corretamente. O erro é computado, o peso dos exemplos classificados corretamente é reduzido e o peso dos exemplos classificados incorretamente é aumentado. O voto de cada classificador individual é ponderado proporcionalmente ao seu desempenho (Wang, Hao, Ma & Jiang 2011).

No método *bagging* (*bootstrap aggregation*), diferentes subconjuntos de treinamento são formados aleatoriamente com reposição a partir do conjunto de treinamento completo. Cada subconjunto de treinamento é usado como entrada para os classificadores base. Todos os classificadores extraídos são combinados usando um voto majoritário. Enquanto o *bagging* pode gerar classificadores em paralelo, *boosting* os gera sequencialmente.

Random Forests é outro método *ensemble*, o qual constrói muitas árvores de decisão que serão usadas para classificar um novo exemplo por meio do voto majoritário. Cada árvore de

decisão usa um subconjunto de atributos selecionados aleatoriamente a partir do conjunto original, contendo todos os atributos. Além disso, cada árvore usa uma amostra *bootstrap* diferente da base de dados da mesma maneira que o *bagging*.

Geralmente, *bagging* é mais preciso do que um único classificador, porém as vezes ele é muito menos preciso do que o *boosting*. Por outro lado, o *boosting* pode criar *ensembles* menos precisos do que um único classificador. Além disso, em algumas situações, o *boosting* pode super-ajustar base de dados ruidosas, diminuindo seu desempenho. *Random Forests*, por outro lado, são mais robustas do que *boosting* com relação a ruído; mais rápido do que *bagging* e *boosting*; seu desempenho é tão bom quanto *boosting* e as vezes melhor e não super-ajusta (Breiman 2001).

Neste estudo, há um interesse especial pelo método *Bagging* uma vez que a *Random Forest* utiliza este método para criar as árvores de uma floresta. Dessa forma, o método *Bagging* será detalhado na seção seguinte.

2.1.4 *Bagging*

O método *Bagging* (Breiman 1996) é um dos algoritmos de aprendizado *ensemble* mais antigos. Ele usa amostras *bootstrap* do conjunto de treinamento (Efron 1979). Amostragem *bootstrap* é uma técnica de amostragem com reposição: a partir do conjunto de treinamento inicial, são selecionados aleatoriamente exemplos para um novo subconjunto de treinamento. Assim, no método *Bagging* diferentes subconjuntos são aleatoriamente construídos, com reposição, a partir do conjunto original. Devido à reposição, alguns exemplos podem ser escolhidos mais de uma vez e outros podem não ser escolhidos. A amostragem *bootstrap* mais comumente utilizada é a E_0 , explicada a seguir.

Cada subconjunto gerado possui o mesmo tamanho (número de exemplos) do conjunto original. Considerando um conjunto de treinamento T com n exemplos, T_k é uma amostra *bootstrap* do conjunto de treinamento a partir de T com reposição, contendo n exemplos. Cada subconjunto T_k é usado para treinar um classificador diferente $\{h_k(x)\}$. A estratégia de combinação dos classificadores é o voto majoritário.

Para uma dada amostra T_k , um exemplo do conjunto de treinamento T tem uma probabilidade $1 - (1 - \frac{1}{n})^n$ de ser selecionado pelo menos uma vez entre os n exemplos selecionados

aleatoriamente. Para um valor grande de n , isto é aproximadamente $1 - 1/e \approx 63,2\%$, o que significa que cada amostra *bootstrap* contém, em média, somente cerca de 63,2% dos exemplos do conjunto de treinamento original T , com muitos exemplos de treinamento aparecendo múltiplas vezes (Bauer & Kohavi 1999; Dietterich 2000). Os exemplos que não foram selecionados na amostra T_k formam o conjunto *out-of-bag*, ou seja, são os exemplos que não foram utilizados para o treino na construção do classificador h_k .

Assim, a avaliação *out-of-bag* funciona da seguinte maneira: dado um conjunto de treinamento específico T , geram-se conjuntos de treinamento *bootstrap* T_k , constroem-se classificadores $\{h_k(x)\}$ e os deixam votarem para criar o classificador *bagged*. Para cada (x, y) no conjunto de treinamento, agregam-se os votos somente sobre os classificadores para os quais T_k não contém (x, y) . Este é o estimador *out-of-bag*, que fornece uma estimativa para o erro de generalização (Breiman 2001).

Na Figura 2.7 são ilustrados os métodos *Bagging*(a) e *Random Forest*(b), o qual será explicado na Seção 2.1.5 juntamente com a diferença entre ambos (parte tracejada).

Segundo (Acuna & Rojas 2001; Breiman 1996; Freund & Schapire 1996), *Bagging* é muito eficaz quando os classificadores utilizados possuem um comportamento instável (como por exemplo, as árvores de decisão). Um classificador é conhecido como instável quando pequenas mudanças no conjunto de treinamento podem causar grandes mudanças no classificador gerado. Nestes casos, um único classificador instável não é capaz de oferecer uma resposta confiável ao contrário de um conjunto de classificadores, uma vez que um classificador composto pode ter maior chance de acerto (Lopes 2007).

2.1.5 *Random Trees & Random Forests*

Considerando um conjunto de treinamento T com a atributos e n exemplos, seja T_k uma amostra *bootstrap* do conjunto de treinamento a partir de T com reposição, contendo n exemplos e usando m atributos aleatórios ($m \leq a$) em cada nó das árvores.

Random Tree é uma árvore induzida aleatoriamente a partir de um conjunto de árvores possíveis, usando m atributos aleatórios em cada nó. O termo “aleatoriamente” significa que cada árvore tem uma chance igual de ser amostrada. *Random Trees* podem ser geradas eficientemente e a combinação de grandes conjuntos de *Random Trees* geralmente leva a

Figura 2.7: Funcionamento dos métodos Bagging (a) e Random Forest (b)

modelos precisos (Zhao & Zhang 2008; Dubath, Rimoldini, Süveges, Blomme, López, Sarro, De Ridder, Cuypers, Guy, Lecoecur, Nienartowicz, Jan, Beck, Mowlavi, De Cat, Lebzelter & Eyer 2011).

Random Forest é definido formalmente como segue (Breiman 2001): é um classificador composto por uma coleção de árvores $\{h_k(x)\}$, $k = 1, 2, \dots, L$, onde T_k são amostras aleatórias independentes e identicamente distribuídas e cada árvore vota na classe mais popular para a entrada x .

Como já foi mencionado, *Random Forests* aplicam o mesmo método que o *bagging* para produzir amostras aleatórias de conjuntos de treinamento (amostras *bootstraps*) para cada *Random Tree*. Cada novo conjunto de treinamento é construído, com reposição, a partir do conjunto de treinamento original. A árvore é construída usando este novo subconjunto e uma seleção aleatória de atributos. A cada nó da árvore, um subconjunto de m atributos é selecionado aleatoriamente e avaliado. O melhor atributo é, então, escolhido para dividir o nó. O valor m é fixado para todos os nós. As árvores crescem sem poda. Na Figura 2.7 (b) é ilustrado o funcionamento de uma *Random Forest* que diferencia do funcionamento

do *bagging* na parte tracejada da figura, uma vez que na *Random Forest* as árvores são construídas selecionando aleatoriamente m atributos em cada nó das árvores, o que não acontece no *bagging* em que todos os a atributos são usados.

Breiman (2001) justifica o uso do método *bagging* em *Random Forests* por duas razões: o uso do *bagging* parece melhorar o desempenho quando atributos aleatórios são usados; *bagging* pode ser usado para fornecer estimativas contínuas do erro de generalização do conjunto combinado de árvores, assim como estimativas para força e correlação, usando o estimador *out-of-bag*.

O erro de classificação da floresta depende da força das árvores individuais da floresta e da correlação entre quaisquer duas árvores na floresta (Breiman 2001; Breiman & Cutler 2004; Breiman 2004; Ma, Guo & Cukic 2007), a saber:

- Correlação entre as árvores da floresta: duas medidas de aleatoriedade (uso do *bagging* e seleção aleatória de atributos) fazem com que as árvores sejam diferentes e, portanto, diminui a correlação entre elas. A baixa correlação tende a diminuir a taxa do erro de classificação.
- Força da árvore individual na floresta: pode ser interpretada como uma medida de desempenho para cada árvore. Uma árvore com uma taxa de erro baixa é um classificador forte. Assim, aumentando a força das árvores individuais, reduz-se a taxa de erro da floresta.

2.2 Considerações Finais

Neste capítulo foram apresentados alguns conceitos utilizados durante o desenvolvimento deste projeto de mestrado, como por exemplo, aprendizado de máquina, *Random Forest* e Árvores de Decisão. Foi descrito como uma *Random Forest* funciona e como ela é construída. No capítulo seguinte são apresentados conceitos de biologia envolvendo expressão gênica.

Conceitos Biológicos

Neste capítulo serão apresentados alguns conceitos biológicos utilizados no projeto, como expressão gênica e *microarrays*.

3.1 Biologia Molecular - Conceitos Básicos

Durante o desenvolvimento do presente trabalho, foram utilizados dados de expressão gênica obtidos por meio da técnica conhecida como *microarray*. Assim, esta seção traz alguns conceitos importantes para o entendimento de tal técnica.

3.1.1 DNA

Uma molécula de DNA (ácido desoxirribonucleico) consiste em duas longas fitas anti-paralelas entrelaçadas em forma de dupla hélice, compostas por uma sequência de nucleotídeos. Os nucleotídeos são compostos de açúcares pentoses (com cinco carbonos), um ou mais grupos fosfatos ligados aos carbonos e uma base contendo nitrogênio. O açúcar dos nucleotídeos de DNA é uma desoxirribose (Figura 3.1) ligada a um grupo fosfato e a base nitrogenada pode ser adenina, citosina, guanina ou timina (Alberts, Johnson, Lewis, Raff,

Figura 3.1: *Desoxirribose. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)*

Roberts & Walter 2010), conforme ilustrado na Figura 3.2. Para simplificar a nomenclatura dos nucleotídeos é utilizada a abreviação da base nitrogenada que o identifica, ou seja, nucleotídeos com a base nitrogenada adenina são abreviados pela letra (A), citosina (C), guanina (G) e timina (T). As bases nitrogenadas podem ser de dois tipos: purinas, com um anel duplo (adenina e guanina) e pirimidinas, com um anel simples (citosina e timina) (Klug, Cummings, Palladino & Spencer 2010), conforme ilustrado na Figura 3.3.

As bases dos nucleotídeos mantêm as duas fitas de DNA unidas por meio de pontes de hidrogênio entre elas. Segundo o modelo de Watson-Crick (Watson & Crick 1953), a base (A) pareia somente com a base (T) e a base (C) pareia somente com a base (G). A adenina forma duas pontes de hidrogênio com a timina e a guanina forma três pontes de hidrogênio com a citosina. Esse pareamento específico das bases $A = T$ e $G \equiv C$ resulta em uma complementariedade das bases, ou seja, as fitas de DNA são complementares (Klug, Cummings, Palladino & Spencer 2010). Assim, ao saber a sequência de uma das fitas, automaticamente, sabe-se a sequência da sua fita complementar.

A fita de DNA possui uma polaridade indicada por uma extremidade 3' e por uma extremidade 5'. Na extremidade 3' há um grupo hidroxil livre ligado ao carbono número 3 da pentose e na extremidade 5' há um fosfato livre ligado ao carbono número 5 da pentose. A união dos nucleotídeos é feita por meio de uma ligação fosfodiéster entre a extremidade 3' de um nucleotídeo e a extremidade 5' de outro (Klug, Cummings, Palladino & Spencer 2010) (Figura 3.4). Assim, o termo anti-paralelas significa que enquanto uma fita corre no sentido 5' para 3', a outra fita vai no sentido 3' para 5'. Uma estrutura de DNA, ilustrando todas as características citadas anteriormente, é mostrada na Figura 3.5.

Uma sequência de nucleotídeos no DNA armazena a informação genética. Os segmentos de DNA que contém a informação genética são denominados genes. A série completa de

Figura 3.2: Nucleotídeo de DNA. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)

Figura 3.3: Bases nitrogenadas do DNA. Adaptado de (Klug, Cummings, Palladino & Spencer 2010)

Figura 3.4: Ligação fosfodiéster. Adaptado de (Klug, Cummings, Palladino & Spencer 2010)

Figura 3.5: *Estrutura de DNA. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)*

informações genéticas contidas no DNA é denominada genoma. O genoma contém informações para todas as proteínas e moléculas de RNA que o organismo irá sintetizar (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010). As proteínas possuem diversas funções biológicas (Lewis 2001), entre elas pode-se citar a função estrutural e a função regulatória. O fluxo da informação genética ocorre da seguinte maneira: a partir do DNA é formado o RNA (processo denominado transcrição) e o RNA é utilizado como molde para a síntese de proteínas (processo denominado tradução). Este princípio ($\text{DNA} \rightarrow \text{RNA} \rightarrow \text{proteína}$) é conhecido como dogma central da biologia molecular (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010; Klug, Cummings, Palladino & Spencer 2010).

3.1.2 RNA

A molécula de RNA (ácido ribonucleico) também é composta por uma sequência de nucleotídeos unidos entre si por meio de ligações fosfodiéster. Porém, o RNA possui algumas diferenças do DNA (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010; Klug, Cummings, Palladino & Spencer 2010):

- o açúcar dos nucleotídeos do RNA é ribose em vez de desoxirribose, conforme ilustrado na Figura 3.6;
- as bases nitrogenadas do RNA podem ser adenina (A), guanina (G) e citosina (C), assim como no DNA, porém o RNA contém a base uracila (U), em vez da timina (T). A estrutura da base uracila é apresentada na Figura 3.7. Nesse caso, a base adenina

Figura 3.6: *Ribose. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)*

Figura 3.7: *Base nitrogenada Uracila. Adaptado de (Klug, Cummings, Palladino & Spencer 2010)*

pareia com a base uracila e a base guanina pareia com a base citosina (assim como no DNA);

- a estrutura do RNA é uma fita simples;

Existem três classes principais de RNA: RNA mensageiro (mRNA), RNA transportador (tRNA) e RNA ribossômico (rRNA). O mRNA é uma molécula de fita simples, produzida a partir de um filamento molde de DNA, em um processo conhecido como transcrição. O mRNA tem como função transmitir a informação genética do núcleo para o citoplasma, pois os genes estão localizados no núcleo celular e os ribossomos (responsáveis pela síntese de proteínas) localizam-se no citoplasma. A sequência de um mRNA é lida em grupos de três nucleotídeos por vez. Cada uma dessas trincas de nucleotídeos (denominadas códons) codificam um único aminoácido (subunidade que forma as proteínas). Diversos códons codificam um mesmo aminoácido, conforme é mostrado na Tabela 3.1.

O tRNA é uma molécula que carrega os aminoácidos para o ribossomo durante a tradução. Cada tRNA liga-se em uma extremidade a um aminoácido específico, possuindo em sua outra extremidade uma sequência de três nucleotídeos denominada anticódon. O anticódon, por meio do pareamento de bases, habilita o tRNA a reconhecer um códon (ou um grupo

Tabela 3.1: *Os 20 aminoácidos com seus respectivos símbolos, abreviaturas e os códons que os codificam*

Símbolo	Abreviação	Códons
A	Ala	GCU, GCG, GCC, GCA
R	Arg	CGU, CGG, CGC, CGA , AGG, AGA
D	Asp	GAU, GAC
N	Asn	AAU, AAC
C	Cys	UGU, UGC
E	Glu	GAG, GAA
Q	Gln	CAG, CAA
G	Gly	GGU, GGG, GGC, GGA
H	His	CAU, CAA
I	Ile	AUU, AUC, AUA
L	Leu	CUU, CUG, CUC, CUA, UUG, UUA
K	Lys	AAG, AAA
M	Met	AUG
F	Phe	UUU, UUC
P	Pro	CCU, CCG, CCC, CCA
S	Ser	UCU, UCG, UCC, UCA, AGU, AGC
T	Thr	ACU, ACG, ACC, ACA
W	Trp	UGG
Y	Tyr	UAU, UAC
V	Val	GUU, GUG, GUC, GUA

de códons) no mRNA, conforme ilustrado na Figura 3.8.

O rRNA juntamente com as proteínas ribossomais (mais de 50 proteínas diferentes) formam os ribossomos. O ribossomo agarra-se à porção terminal de um mRNA e desloca-se ao longo desse mRNA capturando moléculas de tRNA, formando assim uma nova cadeia de proteína por meio da ligação dos aminoácidos transportados pelos tRNAs (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010). Os ribossomos são formados por duas subunidades de tamanhos diferentes, conhecidas como subunidades maior e menor. Estas subunidades interligam-se para sintetizar as proteínas e possuem papel fundamental nos sinais de início da tradução.

3.1.3 Expressão gênica

Expressão gênica é o processo pelo qual as sequências de nucleotídeos dos genes são interpretados em aminoácidos e, assim, em proteínas. Tal processo é ilustrado na Figura 3.9.

Figura 3.8: Molécula de tRNA ligada a um códon no mRNA. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)

A expressão gênica é composta por duas etapas: transcrição e tradução.

A transcrição (primeiro passo da expressão gênica) é o processo no qual um RNA é formado a partir de uma fita-molde de DNA. Esse processo pode ser subdividido em três fases: iniciação, alongamento e término (Klug, Cummings, Palladino & Spencer 2010).

O DNA possui uma região denominada promotora. Essa região é uma sequência especial de nucleotídeos que indica o início para a síntese de RNA. Esta região é reconhecida pela RNA-polimerase e outras moléculas utilizadas na transcrição, denominadas fatores gerais de transcrição. A RNA-polimerase é a enzima que realiza a transcrição, por meio da catalisação da formação de ligações fosfodiéster que conectam os nucleotídeos formando assim uma molécula de RNA. A RNA-polimerase move-se sobre o DNA, abrindo a dupla hélice e, assim, expondo uma nova região da fita-molde para o pareamento de bases por complementariedade. A molécula de RNA é estendida na direção 5' para 3'. Os fatores gerais de transcrição são proteínas que ajudam a posicionar corretamente a RNA-polimerase sobre o promotor. Sem estes fatores, a RNA-polimerase não consegue ligar-se diretamente aos sítios promotores e iniciar a transcrição. Além disso, esses fatores auxiliam na separação das duas fitas de DNA para que a transcrição inicie e liberam a RNA-polimerase do promotor, uma vez que a transcrição tenha iniciado.

Figura 3.9: *Expressão gênica. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)*

Os núcleos eucarióticos possuem três tipos de RNA-polimerases: RNA-polimerase I, RNA-polimerase II e RNA-polimerase III. Os três tipos são estruturalmente similares, porém transcrevem direntes tipos de genes. As RNA-polimerases I e III transcrevem os genes que codificam tRNA, rRNA e diversos pequenos RNAs. Já a RNA-polimerase II transcreve a grande maioria dos genes, inclusive os que codificam mRNA e consequentemente, as proteínas. Sendo assim, será utilizada a RNA-polimerase II para explicar o conceito de expressão gênica.

A iniciação ocorre quando a RNA-polimerase II e os fatores gerais de transcrição reconhecem e ligam-se à região promotor para formar um complexo de iniciação de transcrição. Após a formação deste complexo sobre o DNA, a RNA-polimerase deverá ter acesso à fita molde no ponto inicial da transcrição. Então, um dos fatores de transcrição é responsável pela abertura do DNA e assim, pela exposição da fita molde. A RNA-polimerase mantém-se no promotor até sofrer uma série de alterações estruturais que permitem sua saída do promotor e a entrada na fase de extensão (alongamento) da transcrição (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010).

Após isso, começa a etapa de alongamento e desenrolamento da dupla hélice. A RNA-

polimerase desliza ao longo da molécula de DNA, separando por um momento as duas fitas, onde uma delas servirá de molde para a formação da molécula de mRNA correspondente. Após a transcrição, a molécula de DNA volta ao seu estado original, ou seja, a dupla hélice.

Por fim, o processo termina quando a RNA polimerase II reconhece uma sequência de terminação específica no DNA.

Em células eucarióticas, o RNA transcrito é chamado de pré-mRNA ou RNA transcrito primário. Tal transcrito passa por um processamento constituído de três etapas:

- Capeamento na extremidade 5': ocorre a adição de uma estrutura (*cap* 5'). O *cap* é adicionado antes mesmo que o transcrito primário esteja completo, ou seja, assim que a RNA-polimerase II produziu aproximadamente 25 nucleotídeos de RNA, a extremidade 5' da nova molécula de RNA é modificada pela adição do *cap*. Este *cap* ajuda a célula a distinguir os mRNAs dos outros tipos de moléculas de RNA, como por exemplo, as moléculas produzidas pelas RNA-polimerases I e III. Além disso, este capeamento ajuda o RNA a ser adequadamente processado e exportado (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010);
- Poliadenilação na extremidade 3': a extremidade 3' do transcrito primário é clivada enzimaticamente e então, ocorre a poliadenilação por adição de uma cauda de poli-A (aproximadamente 200 nucleotídeos A) (Klug, Cummings, Palladino & Spencer 2010; Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010). Na ausência dessa cauda, os transcritos de RNA são rapidamente degradados (Klug, Cummings, Palladino & Spencer 2010);
- *Splicing*: remoção dos íntrons (sequências não codificantes) e união dos éxons (sequências codificantes), conforme ilustrado na Figura 3.10. Uma vantagem do *splicing* é que ele permite que um mesmo gene produza um grupo de diferentes proteínas. Isso acontece devido às diferentes maneiras de *splicing* que os transcritos de muitos genes sofrem.

Somente após ter ocorrido o *splicing* e processamento das extremidades 5' e 3', o pré-mRNA será denominado mRNA (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010).

Figura 3.10: *Splicing do RNA. Adaptado de (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010)*

Após o processamento, o mRNA torna-se maduro e é, então, transportado para o citoplasma, para que ocorra a tradução.

Na tradução, é realizada a síntese de proteína a partir de um mRNA, em associação aos ribossomos. No processo de tradução, são utilizados os tRNAs, os quais servem de molécula adaptadora entre um códon de mRNA e o aminoácido correspondente. Assim, a sequência de nucleotídeos do mRNA é lida de uma extremidade à outra de três em três nucleotídeos.

O processo de tradução, assim como o processo de transcrição, pode ser dividido em três fases: iniciação, alongamento e terminação (Klug, Cummings, Palladino & Spencer 2010). O início da tradução ocorre quando uma subunidade ribossomal menor se liga a uma molécula de mRNA em um códon de iniciação (AUG), o qual é reconhecido por um tRNA especial (tRNA iniciador) que sempre carrega o aminoácido metionina. Uma subunidade ribossomal maior se liga para completar o ribossomo e iniciar a síntese proteica. A subunidade menor fornece uma região sobre a qual os tRNAs podem ser eficientemente pareados sobre os códons de um mRNA e a subunidade maior catalisa a formação das ligações peptídicas entre os aminoácidos, formando uma cadeia polipeptídica (proteína) (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010).

Uma vez que a síntese foi iniciada, cada novo aminoácido é adicionado à cadeia em formação em um ciclo de quatro passos principais: ligação do tRNA, formação da ligação peptídica, translocação das subunidades maior e menor. Como resultado dos dois passos de translocação (subunidades maior e menor), o ribossomo completo move-se três nu-

cleotídeos sobre o mRNA e é posicionado para dar início ao próximo ciclo (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010). Esse ciclo é repetido cada vez que um aminoácido é adicionado à cadeia polipeptídica.

A molécula de mRNA é lida códon a códon, na direção 5' para 3', até alcançar um dos três códons de terminação (UAA, UAG ou UGA). Estes códons não são reconhecidos por um tRNA e portanto, não determinam um aminoácido. Assim, os códons de terminação sinalizam para o ribossomo o final da tradução. Então, proteínas conhecidas como fatores de liberação ligam-se aos ribossomos, finalizando a tradução e liberando o polipeptídeo completo. O ribossomo, então, libera o mRNA e separa-se nas duas subunidades maior e menor (Alberts, Johnson, Lewis, Raff, Roberts & Walter 2010).

Técnicas

A análise da expressão gênica pode fornecer informações importantes sobre as funções de uma célula (Alberts 1997). Existem diversas técnicas para obter a expressão dos genes como, por exemplo, SAGE (*Serial Analysis of Gene Expression*), MPSS (*Massively Parallel Signature Sequence technology*) e *microarray* (de Souto, Lorena, Delbem & de Carvalho 2003). Esta última técnica merece um destaque entre as demais pois as bases utilizadas no desenvolvimento desta pesquisa são bases de expressão gênica por *microarrays*.

A técnica SAGE (*Serial Analysis of Gene Expression*) (Velculescu, Zhang, Vogelstein & Kinzler 1995) é baseada principalmente em dois princípios (Yamamoto, Wakatsuki, Hada & Ryo 2001): primeiro, uma pequena sequência de cDNA (DNA complementar) de 9-10 pares de bases (*tag* ou etiqueta) possui informações suficientes para a identificação de um único transcrito; segundo, múltiplos *tags* podem ser concatenados (ligados) e sequenciados, revelando a sequência de milhares de *tags* simultaneamente, ou seja, a concatenação possibilita a análise dos transcritos de um modo serial. Essa análise serial dos *tags* de sequência necessita de mecanismos para a identificação dos limites de cada *tag*.

Tanto problemas técnicos como problemas intrínsecos ao método SAGE são destacados para a técnica de SAGE. Como problemas técnicos destacam-se a desvantagem da necessidade de uma quantidade relativamente alta de mRNA e dificuldade relativa em construir bibliotecas de *tags* (Yamamoto, Wakatsuki, Hada & Ryo 2001). Já entre os problemas in-

trínsecos ao método SAGE destaca-se o comprimento do *tag* é extremamente curto (9-10 pares de bases) e isso dificulta novas análises, especialmente quando os *tags* são derivados de genes não conhecidos (Yamamoto, Wakatsuki, Hada & Ryo 2001).

MPSS (*Massively Parallel Signature Sequence technology*) (Brenner, Johnson, Bridgham, Golda, Lloyd, Johnson, Luo, McCurdy, Foy, Ewan, Roth, George, Eletr, Albrecht, Vermaas, Williams, Moon, Burcham, Pallas, DuBridge, Kirchner, Fearon, Mao & Corcoran 2000) é uma técnica baseada na produção de sequências curtas de 16-20 bases adjacentes ao sítio de restrição da enzima *DpnII* mais próximo à extremidade 3' do mRNA (Liu). Neste método, há uma combinação de clonagem e amplificação de cDNA na superfície de milhares de *microbeads* e uma elevada capacidade de sequenciamento de DNA não baseada em gel. A sua desvantagem é o elevado custo, pois é uma tecnologia baseada em instrumentos sofisticados e protegida por direitos autorais.

A técnica de *microarray* merece um destaque entre as demais pois as bases utilizadas no desenvolvimento desta pesquisa são bases de expressão gênica por *microarrays*. Assim, tal técnica será descrita em maiores detalhes na seção seguinte.

3.1.4 *Microarrays*

A tecnologia de *microarrays*, ou microarranjos de DNA, é usada para examinar a expressão de milhares de genes em diferentes tecidos de um determinado organismo de maneira simultânea (Guindalini & Tufik 2007). Essa tecnologia pode ser utilizada também para o estudo da genômica funcional (Lockhart, Dong, Byrne, Follettie, Gallo, Chee, Mittmann, Wang, Kobayashi, Norton & Brown 1996; Sharp, Xu, Lit, Walker, Apperson, Gilbert, Glau-ser, Wong, Hershey, Liu, Pinter, Zhan, Liu & Ran 2006), o qual busca identificar variações na expressão de determinados genes, as quais podem ser respostas biológicas naturais de uma patologia, ou alguma outra condição experimental (Cheung, Morley, Aguilar, Massimi, Kucherlapati & Childs 1999).

A maior parte desta tecnologia consiste em uma lâmina de microscopia, na qual moléculas de DNA de fita simples são aplicadas de forma ordenada, isto é, em um arranjo. O preparo de um *microarray* começa pela extração de mRNA. Então, geralmente é feita uma transcrição reversa do mRNA, sintetizando assim cDNA marcado com nucleotídeos identi-

ficados por fluorescência. O cDNA (DNA complementar) é um DNA sintetizado a partir de uma molécula de mRNA, cujos íntrons (sequências não codificantes) já foram removidos. Os cDNAs são então desnaturados e incubados com o *microarray* para hibridizar com os seus pontos, os quais contêm as sequências complementares de DNA. Após isso, o *microarray* é lavado e depois rastreado com um *laser* que faz fluorescer o cDNA hibridizado. Os pontos fluorescentes revelam quais os genes que se expressam em um determinado tecido e o seu nível relativo de expressão, uma vez que quanto mais brilhante o ponto, mais o respectivo mRNA se expressa naquele tecido (Klug, Cummings, Palladino & Spencer 2010). Tal processo é ilustrado na Figura 3.11. Nesta figura, o primeiro passo ilustrado é o isolamento do mRNA das amostras de interesse, por exemplo uma amostra de uma pessoa normal (Amostra A) e de uma pessoa com câncer (Amostra B). A partir desses mRNAs, é realizada uma transcrição reversa, sintetizando cDNAs (DNAs complementares) - passo 2. Tais cDNAs são marcados por meio de nucleotídeos identificados por fluorescência. Assim, os cDNAs referentes à uma amostra, por exemplo Amostra A, são marcados pela cor vermelha e os cDNAs referentes à outra amostra, Amostra B, são marcados pela cor verde (passo 3). Os cDNAs marcados são desnaturados e incubados com o *microarray* para que ocorra a hibridização com os pontos do *microarray* onde encontram-se as sequências complementares de DNA, como é ilustrado no passo 4. Por fim, o *microarray* passa por um laser para fluorescer o cDNA que hibridizou (passo 5). Portanto, os pontos que apresentam a cor vermelha, são pontos onde somente o cDNA marcado com a cor vermelha (Amostra A) hibridizou, ou seja, o gene referente àquele ponto só é expresso na Amostra A; os pontos com a cor verde são pontos onde somente o cDNA marcado com a cor verde (Amostra B) hibridizou, ou seja, o gene referente àquele ponto só é expresso na Amostra B; e os pontos com cor amarela são pontos onde tanto o cDNA marcado com a cor vermelha quanto o cDNA marcado com a cor verde hibridizaram, ou seja, o gene referente àquele ponto é expresso tanto na Amostra A quanto na Amostra B. Dessa maneira, a partir dos pontos que fluoresceram é possível determinar quais genes são expressos nas duas amostras de interesse. A intensidade da fluorescência indica o nível relativo de expressão, ou seja, quanto mais brilhante for o ponto, mais expresso é o respectivo mRNA naquele tecido (Klug, Cummings, Palladino & Spencer 2010).

Uma vez que os *microarrays* permitem o estudo de milhares de genes ao mesmo tempo,

Figura 3.11: *Microarray*

eles podem gerar uma grande quantidade de dados sobre expressão gênica. Assim, programas de computador que analisem tais dados são essenciais para organizar o perfil de dados de expressão gênica (Klug, Cummings, Palladino & Spencer 2010), que é o foco do capítulo seguinte.

3.2 Considerações Finais

Neste capítulo foram fornecidos alguns conceitos a respeito de *microarrays* e expressão gênica. No capítulo seguinte será apresentada a metodologia proposta para a geração de uma única árvore a partir de uma *Random Forest*.

Proposta Metodológica

Neste capítulo são descritos alguns trabalhos relacionados encontrados na literatura. A seguir, é descrita a metodologia utilizada na realização do presente projeto, juntamente com os resultados de um experimento preliminar utilizando a metodologia proposta.

4.1 Trabalhos Relacionados

Uma vez que *Random Forests* são eficientes, multi-classe e capazes de lidar com um amplo espaço de atributos, elas têm sido amplamente utilizadas em diversos domínios, inclusive na bioinformática, por exemplo em (Wu, Abbott, Fishman, McMurray, Mor, Stone, Ward, Williams & Zhao 2003; Lee, Park & Song 2005; Díaz-Uriarte & de Andrés 2006; Goldstein, Hubbard, Cutler & Barcellos 2010).

Em (Wu, Abbott, Fishman, McMurray, Mor, Stone, Ward, Williams & Zhao 2003) são comparados resultados obtidos com vários classificadores para distinguir entre pacientes com câncer de ovário e indivíduos normais baseados em dados de espectrometria de massa (MS) obtidos em amostras de soro. Foi utilizada a taxa de erro como medida de desempenho. Concluiu-se que o método da *Random Forest* conduz a uma taxa de classificação errada mais

baixa. Além disso, as análises preliminares sugerem que *Random Forest* e métodos similares a ela podem ser mais úteis do que outros métodos para classificar exemplos baseados em bases de MS.

O trabalho de Lee, Park & Song (2005) analisou o desempenho de classificadores desenvolvidos (entre eles a *Random Forest*) em bases de *microarray* e forneceu orientações para encontrar as ferramentas de classificação mais apropriadas em diversas situações. Foram utilizadas 7 bases de dados, entre elas Lymphoma, uma relacionada com câncer de pulmão e uma de leucemia. Também foi utilizada a taxa de erro como medida de desempenho. A conclusão obtida foi que classificadores como *bagging*, *boosting* e *Random Forest* melhoram o desempenho de CART (árvore de classificação) significativamente e que a *Random Forest* é o melhor método dentre os três quando o número de classes é moderado.

Em (Díaz-Uriarte & de Andrés 2006) foi investigado o uso de *Random Forest* para classificar bases de dados de *microarray* (incluindo problemas multi-classes) e foi proposto um novo método de seleção de genes em problemas de classificação baseados na *Random Forest*. Foi analisado o desempenho da *Random Forest* alterando seus parâmetros, como o número de árvores a serem criadas, número de atributos selecionados aleatoriamente e número mínimo de exemplos em cada folha. Foi utilizado um número mínimo de exemplos entre 1 e 5, número de árvores igual a 1000, 2000, 5000, 10000, 20000 e 40000 e diversos números de atributos selecionados aleatoriamente. Foi concluído que mudanças nestes parâmetros apresentam, na maioria das vezes, efeitos insignificantes, sugerindo que os valores *default* são boas opções. Além disso, concluiu-se que o tempo de execução aumenta linearmente com o número de árvores e a mudança no número mínimo de exemplos entre 1 e 5 tem efeitos insignificantes e assim, seu valor *default* de 1 é apropriado. Foi utilizada a taxa de erro *out-of-bag*. A conclusão geral obtida é que devido ao seu desempenho, *Random Forest* e seleção de genes usando *Random Forest* deveriam tornar-se parte da “caixa de ferramentas padrão” dos métodos para predição de classes e seleção de genes utilizando bases de *microarray*.

Em (Goldstein, Hubbard, Cutler & Barcellos 2010) foi apresentada uma das primeiras ilustrações de sucesso analisando bases de genoma de associação ampla (GWA) com um algoritmo de aprendizado de máquina (*Random Forests*) usando uma base de esclerose múltipla (MS). Foram construídas florestas com 50, 250, 500, 1000 e 2000 árvores e número de

atributos selecionados aleatoriamente igual a 10% do número de atributos. Foi concluído que 10% do número de atributos parece minimizar a taxa do erro *out-of-bag*, porém não muito. A convergência parece ocorrer entre 200-400 árvores. Tal resultado é interessante, uma vez que possui a mesma ordem de grandeza que os resultados obtidos em nossas pesquisas (64-128 árvores) em (Oshiro, Perez & Baranauskas 2012). Foi concluído que a *Random Forest* é computacionalmente viável para bases de GWA e os resultados obtidos fazem sentido biológico baseados em estudos anteriores. Mais importante, novos genes foram identificados como potencialmente associados com MS, sugerindo assim novas direções de investigação para esta doença complexa.

Não foi encontrado nenhum trabalho na literatura com um tema semelhante ao aqui proposto, usando *Random Forest*. Porém, em (Aziz & Ahmed 2011), foi introduzido um novo método para gerar um classificador ótimo e preciso a partir de árvores para várias amostras da mesma base de dados. Foi utilizada a métrica ganho de informação para selecionar o nó que irá fazer parte da nova árvore e as árvores foram geradas por meio do método *bagging*. É criado então um mapa contendo todos os nós utilizados por todas as árvores em todos os níveis, ou seja, do nó raiz ao nó folha. Para cada nível, é calculado o ganho de informação de cada atributo utilizado neste nível e o atributo com maior ganho é então utilizado como o nó daquele nível da árvore final. Isto é feito até que a árvore final tenha sido criada. Foi concluído que dessa maneira consegue-se criar uma única árvore e pode-se verificar os resultados facilmente usando este único classificador ao invés de vários. No estudo, contudo, não há informações sobre o desempenho da árvore criada ou até mesmo comparações entre a árvore final e outros classificadores.

Na seção seguinte é descrito o método proposto nesta pesquisa de mestrado para a construção de uma única árvore.

4.2 Proposta para construção de uma única árvore

Nesta seção é apresentada a proposta para a construção de uma única árvore a partir das árvores de uma *Random Forest*. Foi utilizado o software de código aberto Weka (Hall et al. 2009) para a implementação do algoritmo proposto.

O resultado de uma *Random Forest*, como já foi mencionado, são L árvores. Em um

estudo preliminar deste mestrado, constatou-se que conforme o número de árvores geradas em uma *Random Forest* vai crescendo, tal floresta tende a usar todos os atributos da base de dados (Oshiro, Perez & Baranauskas 2012). Porém isso não é algo útil, uma vez que nem todos os atributos existentes são importantes para a classificação. Assim, foi utilizada uma métrica para identificar quais poderiam ser os melhores atributos e, conseqüentemente, qual seria o atributo a ser escolhido como um nó para compor a árvore final desta pesquisa. A partir deste atributo, novas florestas foram geradas com L árvores para cada subárvore do atributo e o processo foi repetido, como pode ser visto no Algoritmo 1. O parâmetro m foi utilizado com seu valor *default* igual a $\log_2(a)$, pois esta parece ser frequentemente uma boa escolha (Liaw & Wiener 2002).

A construção da árvore proposta, descrita no Algoritmo 1, tem como início a geração de uma *Random Forest* a qual é composta por L *Random Trees* ($RT_1, RT_2, \dots, RT_L$) como indicado na linha 5. Nas linhas 7 a 9 é construído um mapa, o qual contém o nome do atributo utilizado na raiz da k -ésima árvore, $Raiz(RT_k)$, e sua métrica correspondente, $Métrica(Raiz(RT_k))$. Após isso, é construído um mapa final o qual contém o nome de todos os atributos utilizados nas raízes das L árvores e as médias das suas respectivas métricas. Tais médias são calculadas por meio da razão entre a somatória dos valores das métricas do atributo α e o número de vezes que este atributo foi selecionado para ser a raiz de uma árvore, ou seja, o número de vezes que este atributo aparece no *mapa*, como é mostrado nas linhas 10 a 13. A seguir, é escolhido o atributo para compor a árvore final. É selecionado o atributo que apresenta o maior valor da métrica utilizada, conforme mostrado na linha 14. Este atributo irá compor a árvore final (linha 16). Nas linhas 17 a 20 é mostrado como o processo se repete para todos os valores do atributo escolhido, ou seja, para cada valor A_i que o atributo escolhido pode assumir é gerada uma nova base T_i a qual servirá de parâmetro para uma nova *Random Forest*. Na linha 1 é mostrado o ponto de parada do algoritmo, ou seja, a árvore para de crescer quando todos os exemplos de uma determinada base pertencem a uma mesma classe c . Assim, é criado um nó folha, o qual possui a classe c .

Para realizar a construção de uma única árvore a partir das várias árvores de uma *Random Forest* e verificar se esta proposta seria viável, duas questões precisam ser analisadas:

1. a quantidade de árvores a serem construídas em uma floresta. A literatura associada

Algoritmo 1 Construção de uma única árvore usando *Random Forest*

Require: T : um conjunto de n exemplos rotulados $\{(x_i, y_i), i = 1, 2, \dots, n\}$
 L : número de árvores a serem geradas na floresta
 m : número de atributos a serem selecionados aleatoriamente em cada nó de cada árvore
 minNum : número mínimo de exemplos em uma folha

Ensure: $\hat{\text{Árvore}}$: árvore final obtida

procedure construirÁrvore(T, L, m, minNum)

- 1: **if** T contém apenas exemplos de uma única classe c **then**
- 2: $\hat{\text{Árvore}} \leftarrow$ Defina folha com classe c
- 3: **return** $\hat{\text{Árvore}}$
- 4: **end if**
- 5: $\text{RF} \equiv \{\text{RT}_1, \text{RT}_2, \dots, \text{RT}_L\} \leftarrow \text{construirRandomForest}(T, L, m, \text{minNum})$
- 6: $\text{mapa} \leftarrow \text{mapaFinal} \leftarrow \emptyset$
- 7: **for** $k \leftarrow 1$ até L **do**
- 8: $\text{mapa} \leftarrow \text{mapa} \cup \{(\text{Raiz}(\text{RT}_k), \text{Métrica}(\text{Raiz}(\text{RT}_k)))\}$
- 9: **end for**
- 10: **for** cada $(\alpha, \cdot) \in \text{mapa}$ **do**
- 11: Seja $(\alpha, \cdot)$ da forma $\{(\alpha, M_1), (\alpha, M_2), \dots, (\alpha, M_w)\}$ representando os valores da métrica para o atributo α no mapa
- 12: $\text{mapaFinal} \leftarrow \text{mapaFinal} \cup \{(\alpha, \mu = \frac{\sum_{i=1}^w M_i}{w})\}$
- 13: **end for**
- 14: Seja $A \leftarrow \text{argmax}_{\mu | (\alpha, \mu) \in \text{mapaFinal}} \alpha$
- 15: Sejam $A_1, A_2, \dots, A_r$ os possíveis valores do atributo A
- 16: $\hat{\text{Árvore}} \leftarrow$ Defina nó com A como raiz e sejam $S(A_1), S(A_2), \dots, S(A_r)$ os subramos de A
- 17: **for** $i \leftarrow 1$ até r **do**
- 18: Defina $T_i = \{(x, y) \in T \mid A = A_i\}$
- 19: $S(A_i) \leftarrow \text{construirÁrvore}(T_i, L, m, \text{minNum})$
- 20: **end for**
- 21: **return** $\hat{\text{Árvore}}$

fornece poucas informações a respeito de quantas árvores devem ser usadas para constituir uma *Random Forest* com bom desempenho usando a medida AUC. As vezes, aumentar o número de árvores somente aumenta o custo computacional sem obter um ganho significativo no desempenho;

2. a frequência que os atributos são utilizados na raiz das árvores. Atributos irrelevantes não levam somente a um baixo desempenho mas também adicionam dificuldades extras em descobrir conhecimento potencialmente útil (Oh, Lee & Moon 2004; Saeys, Inza & Larrañaga 2007). Dessa maneira, identificar um subconjunto ou um único melhor atributo em uma base de dados de *microarray* pode melhorar o conhecimento humano sobre o tema.

Visando responder essas duas questões, foram realizados dois experimentos neste estudo. No primeiro experimento (Oshiro, Perez & Baranauskas 2012) visou-se analisar o desempenho da *Random Forest* conforme o número de árvores aumenta. Tal experimento buscou por um número (ou um intervalo de números) de árvores a partir do qual não há mais ganho significativo de desempenho, a menos que enormes recursos computacionais estejam disponíveis para grandes bases de dados. Foram construídas *Random Forests* variando o número de árvores das florestas em taxas exponenciais. Foram propostas e utilizadas algumas métricas de densidade para bases de dados, as quais são descritas no Apêndice B.1. A partir de tal experimento, pôde-se concluir que um intervalo entre 64 e 128 árvores é o mais indicado, uma vez que é possível obter um bom balanço entre o AUC e o tempo de processamento. É interessante ressaltar que em (Goldstein, Hubbard, Cutler & Barcellos 2010) foi encontrado um intervalo entre 200 e 400 árvores e foi utilizada como medida de desempenho a taxa de erro, diferentemente da medida aqui utilizada (AUC). Como já mencionado, segundo (Ling, Huang & Zhang 2003) o AUC deveria substituir a precisão (ou taxa de erro, uma vez que esta é igual a $(1 - \text{precisão})$) na comparação de classificadores pois mostrou-se uma medida melhor baseado em extensas comparações experimentais. Os resultados deste experimento são descritos e discutidos mais detalhadamente no Apêndice B.2, tendo sido publicado em (Oshiro, Perez & Baranauskas 2012).

No segundo experimento (Oshiro & Baranauskas 2012), avaliou-se se uma *Random Forest*

utiliza todos os atributos de uma maneira uniforme ou se existem atributos que são mais utilizados que os demais e se tais atributos possuem um bom desempenho. Este experimento foi dividido em três etapas: a primeira etapa visava analisar a frequência que os atributos apareciam na raiz das árvores; a segunda etapa buscava analisar o desempenho dos dez atributos mais utilizados na raiz das árvores usando o erro *out-of-bag* estimado; a terceira etapa visava analisar o desempenho da *Random Forest* usando pré-poda e seu tempo de execução, uma vez que é sabido que as árvores em uma *Random Forest* crescem sem poda mesmo considerando o fato de que a *Random Forest* não superajusta. Além disso, o uso da pré-poda pode acelerar a indução da *Random Forest*. Os resultados deste experimento mostram que a *Random Forest* escolhe um subconjunto de atributos que são mais utilizados nas raízes das árvores geradas. Além disso, o uso do erro *out-of-bag* estimado permite diferenciar os atributos que apresentavam frequências similares e, assim, identificar o melhor atributo usado para compor a raiz de uma árvore. Os resultados deste experimento são descritos e discutidos mais detalhadamente no Apêndice B.3, tendo sido publicado em (Oshiro & Baranauskas 2012).

Utilizando os resultados obtidos nos dois experimentos citados anteriormente, foram realizados experimentos com o Algoritmo 1. No Capítulo 5 são apresentados os resultados do Algoritmo 1 a partir da *Random Forest* comparados com o desempenho de três classificadores: *Random Forest*, J48 (árvore de decisão da Weka (Hall, Frank, Holmes, Pfahringer, Reutemann & Witten 2009), que fornece como saída uma única árvore) e uma árvore gerada de forma aleatória (classificador *Random Tree* da Weka (Hall, Frank, Holmes, Pfahringer, Reutemann & Witten 2009)). Foram utilizadas como medidas comparativas o AUC e o número de nós das árvores. Para avaliar o desempenho, foi realizada validação cruzada com 10-*folds*. As bases utilizadas neste experimento encontram-se descritas no Apêndice A.2.

A avaliação estatística dos resultados é uma parte essencial na validação de novos métodos de aprendizado de máquina (Demšar 2006). O teste estatístico pode ser paramétrico (teste t, ANOVA) ou não paramétrico (Wilcoxon, Friedman). Segundo (Demšar 2006) os testes não paramétricos devem ter preferência em relação aos testes paramétricos, pois eles são mais prováveis a rejeitar a hipótese nula e não correm os riscos de violações das suposições dos testes paramétricos. Tal conclusão foi baseada nas propriedades estatísticas de cada teste,

nos seus pressupostos e no conhecimento dos dados de aprendizagem de máquina.

Assim, a fim de analisar se os resultados dos experimentos reportados no capítulo seguinte são significativamente diferentes, aplicou-se o teste de Friedman (Friedman 1940), considerando um nível de significância de 5%. Se o teste de Friedman rejeita a hipótese nula, é necessário um teste *post-hoc* para verificar em quais pares de classificadores as diferenças são realmente significativas (Demšar 2006). O teste *post-hoc* utilizado foi Benjamini-Hochberg (Benjamini & Hochberg 1995) e foi realizada uma comparação todos *versus* todos, executando todas as possíveis comparações. Os testes foram realizados no software R para computação estatística (<http://www.r-project.org/>).

4.3 Métricas de Similaridade de árvores

Uma vez que o desempenho médio da árvore proposta é conhecido assim como o desempenho médio de outros classificadores, é interessante analisar as árvores obtidas por estes métodos com métricas além do desempenho (taxa de acerto, taxa de erro, AUC, entre outras). Assim, foram utilizadas duas métricas da literatura e propostas outras duas neste trabalho visando comparar o quão similares são as árvores criadas pela metodologia proposta, pelo J48 e pela *Random Tree*, uma vez que não há como comparar com a *Random Forest*, pois essa fornece uma floresta como saída, ou seja, L árvores.

Para calcular três das métricas propostas, foi utilizado como base o Coeficiente de Jaccard (Jaccard 1901). Tal coeficiente analisa a similaridade entre dois conjuntos de amostras e é calculado por meio da seguinte fórmula:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4.1)$$

Em um estudo recente (Perner 2011) foi proposta uma métrica para cálculo da similaridade entre duas árvores. Nesse estudo são realizados cinco passos: primeiro as duas árvores de decisão são transformadas em um conjunto de regras; depois, as regras das duas árvores são ordenadas de acordo com o número *att* de atributos em uma regra; então, as subestruturas de todas as regras são construídas por meio da decomposição das regras em suas subestruturas; a seguir, duas regras i e j das duas árvores de decisão são comparadas; por

último, a medida de similaridade SIM_{ij} é calculada, conforme Equação 4.2.

$$SIM_{ij} = \frac{1}{att}(Sim_1 + Sim_2 + \dots + Sim_k + \dots + Sim_{att}) \quad (4.2)$$

onde $att = \max \{att_i, att_j\}$ e

$$Sim_k = \begin{cases} 1 & \text{se subestrutura for idêntica} \\ 0 & \text{caso contrário} \end{cases}$$

Se a regra contém um atributo numérico (α) então a medida de similaridade é dada por:

$$S = 1 - \frac{\alpha - \alpha'}{t} = 1 - \frac{k_1 - k_1 - |x|}{t} = 1 - \frac{|x|}{t} \quad \text{para } x < t \quad (4.3)$$

$$S = 0 \quad \text{para } x \geq t \quad (4.4)$$

onde $\alpha \leq k_1$ é um atributo numérico, $\alpha' \leq k_2 = k_1 + x$ e t é um valor escolhido pelo usuário para permitir que x esteja em um intervalo de tolerância de $s\%$ de k_1 . Para o desenvolvimento do presente trabalho, foi considerado um valor de s igual a 10%.

Foram utilizadas quatro métricas de similaridade de árvores: Jaccard, Jaccard por nível, Jaccard por nível & teste de Petra e teste de Petra. Dentre as quatro métricas, duas foram reutilizadas da literatura (métricas J e P) e duas foram propostas neste estudo (métricas JN e JNP). Todas as métricas apresentam valores no intervalo $[0, 1]$, sendo que as árvores completamente similares resultam em 1; caso contrário, resultam em zero.

1. Jaccard (J): representa a porcentagem de similaridade entre duas árvores (A e B) em relação aos atributos utilizados por cada uma.
2. Jaccard por nível (JN): representa a similaridade entre duas árvores em relação aos atributos utilizados em cada nível, porém não analisa a topologia da árvore. Aplica-se a fórmula 4.1 em cada nível das árvores, somando os valores e dividindo pelo número total de níveis;

3. Jaccard por nível & teste de Petra (JNP): Mesmo raciocínio da métrica JN, porém quando o atributo é numérico, aplica-se a fórmula 4.3 desenvolvida por (Perner 2011). Assim, se dois atributos numéricos de duas árvores diferentes são iguais em um mesmo nível, ao invés de resultar em 1 (pela fórmula de JN), a similaridade é calculada pela fórmula 4.3;
4. Teste de Petra (P): representa a similaridade topológica entre as árvores.

Como exemplo, considere duas árvores distintas (ilustradas nas Figuras 4.1 e 4.2), geradas por diferentes classificadores e utilizando a mesma base de dados.

Métrica 1: Jaccard

$$J = \frac{2}{4} = 0,5 \quad (4.5)$$

Métrica 2: Jaccard por nível

$$JN = \frac{1 + \frac{1}{3}}{2} = 0,67 \quad (4.6)$$

Métrica 3: Jaccard por nível + teste de Petra $t_1 = 83,1$ e $x_1 = 13$; $t_2 = 5,5$ e $x_2 = 0,5$

$$JNP = \frac{(1 - \frac{13}{83,1}) + (1 - \frac{\frac{0,5}{5,5}}{3})}{2} = \frac{0,84 + 0,30}{2} = 0,57 \quad (4.7)$$

Métrica 4: Teste de Petra $t_1 = 83,1$ e $x_1 = 13$; $t_2 = 5,5$ e $x_2 = 0,5$

$$P = \frac{(1 - \frac{13}{83,1}) + \frac{((1 - \frac{13}{83,1}) + (1 - \frac{0,5}{5,5}))}{2}}{2} = \frac{0,84 + 0,44}{2} = 0,64 \quad (4.8)$$

É possível notar a partir dos exemplos fornecidos que, computacionalmente, a métrica 1 (Jaccard) é a métrica mais simples de ser calculada; a métrica 2 (Jaccard por nível) é mais simples do que as métricas 3 e 4 e a métrica 3 (Jaccard por nível + teste de Petra) é mais simples do que a métrica 4 (Teste de Petra).

4.4 Considerações Finais

Este projeto de mestrado tem como objetivo criar uma única árvore a partir das árvores geradas por uma *Random Forest*. Para atingir tal objetivo, foram realizados dois

Figura 4.1: *Árvore criada pelo classificador J48 utilizando a base DLBCLTumor.*

Figura 4.2: *Árvore criada pelo Algoritmo 1 utilizando a base DLBCLTumor.*

experimentos preliminares, reportados em (Oshiro, Perez & Baranauskas 2012) e (Oshiro & Baranauskas 2012). Uma vez que a árvore final foi criada, comparou-se a sua estrutura à estrutura de outras árvores criadas por outros algoritmos a fim de analisar se ambas árvores são similares e, portanto, fornecem as mesmas informações a respeito de uma determinada base de dados, ou seja, um determinado problema biológico.

Também foram apresentadas neste capítulo a metodologia utilizada neste projeto e as métricas de similaridade entre árvores utilizadas, sendo que duas métricas foram propostas resultantes deste estudo. Foram realizados 2 experimentos que serviram de base para a construção da árvore final. No capítulo seguinte são apresentados os resultados dos experimentos realizados utilizando a metodologia proposta, assim como os resultados das métricas de similaridades.

Resultados & Discussão

Após obter as respostas às questões levantadas na Seção 4.2, achou-se melhor construir florestas visando a estabilidade dos atributos mais utilizados, ou seja, florestas usando a árvores ou $2a$ árvores. Estes valores foram escolhidos com base nos resultados do Experimento 2 descrito na Seção B.3. É importante ressaltar que com o primeiro experimento descobriu-se o número de árvores necessário para obter um bom desempenho da *Random Forest* sem um alto custo computacional. Porém, para se obter uma estabilidade dos atributos utilizados pelas árvores, o número de árvores deve ser maior. Assim, foi realizado o segundo experimento e descobriu-se então, o número de árvores necessário para se obter uma maior estabilidade do conjunto de atributos mais utilizados. Uma vez que com o aumento do número de árvores todos os atributos parecem ser utilizados, é necessário o uso de uma métrica para filtrar os melhores atributos e, assim, não utilizar um atributo que não é importante ou bom para o desempenho da árvore final. Visando isso, foi utilizada a métrica da frequência considerando o erro *out-of-bag* estimado.

A fim de melhorar o erro *out-of-bag* estimado utilizado, um novo experimento foi realizado. Neste novo experimento foi utilizada apenas uma porcentagem dos erros *out-of-bag*

de um determinado atributo. Ou seja, foram descartados os maiores erros *out-of-bag* de um atributo visando assim, eliminar possíveis ruídos com relação à essa métrica. Por exemplo, supondo que um atributo foi utilizado 4 vezes e obteve erros *out-of-bag* de 0.2, 0.4, 0.1 e 0.9. Nesse caso, o valor 0.9 pode ser um ruído, uma vez que este erro elevado pode ser causado pela escolha dos demais atributos da árvore e, portanto, não seria relacionado diretamente ao atributo na raiz. Assim, utilizando apenas uma porcentagem dos erros obtidos (no exemplo, menos de 75%), o valor 0.9 não é levado em consideração.

Foram testados quatro valores de porcentagens: 70%, 80%, 90% e 95%. Os melhores resultados foram obtidos utilizando 70% e 90%. Dessa forma, foram realizados três experimentos utilizando o Algoritmo 1: uso da métrica da frequência considerando o erro *out-of-bag* estimado (100% dos valores obtidos) e considerando uma porcentagem dos erros *out-of-bag* obtidos (70% e 90%).

Procurando analisar melhor o desempenho da árvore proposta, seus resultados foram comparados com outros três classificadores: *Random Forest*, J48 e *Random Tree*. A *Random Forest* foi executada com seus valores *default* e os números de árvores utilizados foram 128, a e $2a$. O valor 128 foi escolhido baseado no Experimento 1, descrito no Apêndice B.2 e a conclusão que a partir desse número o desempenho da *Random Forest* não parece melhorar significativamente. Apesar disso, florestas contendo a e $2a$ árvores foram construídas para uma melhor comparação com a árvore proposta. O classificador J48 foi executado com seus valores *default*, com poda e sem poda. O classificador *Random Tree* também foi executado com seus valores *default*. Foram realizados testes de Friedman para analisar se os resultados obtidos eram significativamente diferentes.

Na Tabela 5.1 estão os valores de AUC obtidos e na Tabela 5.3 são mostrados os números de nós (tamanho) das árvores e florestas criadas em função de $\log_{10}$, já que os números de nós das florestas variam muito. Nas duas tabelas são apresentados os resultados dos seguintes classificadores: Algoritmo 1 utilizando a e $2a$ árvores e 100% dos valores de erro *out-of-bag* (A1-a(100) e A1-2a-(100)); Algoritmo 1 utilizando a e $2a$ árvores e 70% dos valores de erro *out-of-bag* (A1-a(70) e A1-2a-(70)); Algoritmo 1 utilizando a e $2a$ árvores e 90% dos valores de erro *out-of-bag* (A1-a(90) e A1-2a-(90)); J48 com seus valores *default* e com poda (J48) e J48 com seus valores *default* e sem poda (J48(U)); *Random Tree* com seus valores *default*

(RT); *Random Forest* com seus valores *default* e número de árvores igual a 128, a e $2a$ árvores (RF-128, RF- a e RF- $2a$, respectivamente).

Nas Tabelas 5.2 e 5.4 são apresentados os resultados dos testes de Friedman para os valores de AUC e número de nós, respectivamente. Nestas tabelas, Δ ($\blacktriangle$) indica que o classificador na linha especificada é melhor (significativamente) do que o classificador na coluna especificada; ∇ ($\blacktriangledown$) o classificador na linha especificada é pior (significativamente) do que o classificador na coluna especificada; o indica nenhuma diferença. O triângulo inferior destas tabelas não são mostrados pois eles apresentam resultados opostos ao triângulo superior por simetria.

Analisando a Tabela 5.1 pode-se observar que a *Random Forest*, na média, apresentou os melhores valores de AUC na maioria das bases (com exceção das bases DLBCLOutcome, ma-2003-breast e prostate-outcome) e o melhor *rank* médio. Em duas bases (E2A-PBX1-completo e T-ALL-completo) quase todos os classificadores, com exceção da *Random Tree*, apresentaram AUC igual a um. Comparando a metodologia proposta com o J48, J48(U) e a *Random Tree*, pode-se notar que o Algoritmo 1 utilizando 90% dos erros *out-of-bag* e $2a$ árvores (A1-2a(90)) apresentou o melhor *rank* médio dentre os quatro algoritmos. No geral, o Algoritmo 1 apresentou valores de AUC um pouco melhores do que os algoritmos J48 e J48(U) e muito melhores em relação a *Random Tree*. Isso pode ser confirmado ao analisar a Tabela 5.2, uma vez que pode-se notar que os classificadores J48 e J48(U) foram piores, porém não significativamente, do que o Algoritmo 1 no geral (com exceção somente do A1-a(100)) e a *Random Tree* foi significativamente pior do que todos os demais classificadores. Já a *Random Forest* foi significativamente melhor do que os demais classificadores. Analisando o desvio padrão, os algoritmos J48 e J48(U) mostraram-se um pouco mais estáveis do que o Algoritmo 1 no geral, no entanto, o melhor *rank* médio do desvio padrão dentre os quatro algoritmos restantes foi do Algoritmo 1 utilizando 70% dos erros *out-of-bag* e $2a$ árvores (A1-2a(70)). Já a *Random Tree* mostrou-se bem mais instável como esperado.

Na realização do teste de Friedman para os números de nós das árvores foram utilizados somente 9 dos 12 classificadores, para os quais este teste é aplicável. Foram tirados os valores da *Random Forest* uma vez que eles eram valores extremamente altos por se tratar de uma floresta. Assim, tais valores foram desconsiderados a fim de uma melhor comparação entre

os números de nós das árvores criadas por cada classificador.

Analisando a Tabela 5.3, nota-se que a *Random Forest* apresenta o pior desempenho com relação ao número de nós utilizados (tamanho da árvore). Já o J48, o J48(U) e o Algoritmo 1, mais uma vez, apresentam desempenhos similares e melhores do que a *Random Tree* e o melhor *rank* médio dentre os quatro é do Algoritmo 1 utilizando 100% dos erros *out-of-bag* e 2a árvores (A1-2a(100)). Analisando a Tabela 5.4, nota-se que a *Random Tree* foi pior significativamente do que os outros três classificadores. Ao comparar o classificador J48 com o Algoritmo 1, pode-se observar que o J48 foi melhor significativamente em três das sete comparações, pior (não significativamente) em uma das sete e melhor (não significativamente) em duas das sete comparações. Já o algoritmo J48(U) foi melhor significativamente em três das sete comparações e pior (não significativamente) em quatro das sete comparações. Com relação ao desvio padrão, o Algoritmo 1 mostrou-se tão estável quanto o J48 e o J48(U) e apresentou o menor *rank* médio.

5.1 Comparação das árvores

Como descrito na Seção 4.3, foram desenvolvidas algumas métricas para comparação de árvores. Nesta seção, os resultados de tais métricas serão apresentados comparando-se as árvores geradas pelo Algoritmo 1 às árvores geradas pelo algoritmo J48 e pela *Random Tree*. Uma vez que os valores obtidos pelo algoritmo J48 com poda e sem poda foram muito parecidos, utilizou-se as árvores geradas pelo J48 com poda, pois essa é sua configuração *default*. Como mencionado anteriormente, foram testados quatro valores diferentes de porcentagem dos erros *out-of-bag* de um determinado atributo na execução do Algoritmo 1. Nesta seção, entretanto, só foram comparadas as árvores utilizando 90% dos erros *out-of-bag* e florestas construídas com 2a árvores (A1-2a(90)). Essa escolha foi baseada nos resultados do teste de Friedman apresentados na seção anterior.

Na Tabela 5.5, são comparadas as árvores geradas utilizando as 24 bases descritas na Seção A.2 e o Algoritmo 1 (A1-2a(90)) e J48. Na Tabela 5.6, são comparados o Algoritmo 1 (A1-2a(90)) e *Random Tree*. Os valores acima de 0,5 estão destacados em negrito.

Analisando os valores obtidos mostrados na Tabela 5.5 é possível notar que as árvores obtidas utilizando o Algoritmo 1 são, em geral, diferentes das árvores obtidas pelo J48.

Ou seja, as árvores apresentam atributos diferentes em suas estruturas. Pode-se observar, entretanto, que há algumas árvores parecidas, como é o caso das árvores obtidas a partir das bases MLL-completo e T-ALL-completo, uma vez que ambas apresentam valores de métricas acima de 0,5. No caso da base T-ALL, podemos notar que as árvores obtidas pelos dois algoritmos apresentam os mesmos genes nos mesmos níveis, pois as métricas J e JN possuem valores iguais a 1.

Observando os resultados apresentados na Tabela 5.6, pode-se notar que as árvores geradas pelos dois algoritmos (Algoritmo 1 e *Random Tree*) são completamente diferentes, uma vez que a maioria das métricas apresentaram valores iguais a 0. Somente em alguns casos os valores da métrica 1 foram diferentes de zero, porém, o valor mais alto atingido foi 0,02.

Foi calculado o quadrado do coeficiente de correlação (regressão linear) entre as 4 métricas propostas para verificar se havia uma correlação entre elas. Uma vez que as árvores obtidas pelos algoritmos J48 e Algoritmo 1 (A1-2a(90)) apresentaram valores de similaridade maiores do que zero em quase todas as bases, foram utilizados os resultados apresentados na Tabela 5.5 para calcular o coeficiente de correlação entre as métricas. Os coeficientes de correlação R^2 obtidos são apresentados na Tabela 5.7.

Analisando os resultados apresentados na Tabela 5.7 é possível notar que as métricas M_1 e M_2 estão correlacionadas, assim como as métricas M_3 e M_4 . Uma vez que a métrica M_1 é a mais simples de ser calculada (conforme mostrado na Seção 4.3), pode-se sugerir a utilização da métrica M_1 (Jaccard) no lugar da métrica M_2 (Jaccard por nível), pois as duas estão correlacionadas, ou seja, não apresentam resultados muito diferentes. Pode-se também sugerir o uso da métrica M_3 (Jaccard por nível + teste de Petra) no lugar da métrica M_4 (Teste de Petra), uma vez que as duas também estão correlacionadas. Tais correlações podem ser observadas nas Figuras 5.1 e 5.2.

5.2 Considerações Finais

Neste capítulo foram apresentados os resultados obtidos utilizando a metodologia proposta e outros classificadores já conhecidos. Foram também mostradas métricas de similaridade entre as árvores geradas por essa metodologia e as árvores geradas pelos demais classificadores.

Figura 5.1: Resultados das métricas J e JN para as 24 bases.

Figura 5.2: Resultados das métricas JNP e P para as 24 bases.

Tabela 5.1: Valores do AUC para os 12 classificadores, desvio padrão e rank médio.

Base de dados	A1-a(100)	A1-2a(100)	A1-a(70)	A1-2a(70)	A1-a(90)	A1-2a(90)	J48	J48(U)	RF	RF-128	RF-a	RF-2a
welsh-2001-GNF	0,77 (0,07)	0,78 (0,10)	0,78 (0,09)	0,81 (0,12)	0,77 (0,08)	0,75 (0,10)	0,76 (0,11)	0,76 (0,11)	0,76 (0,11)	0,93 (0,06)	0,94 (0,06)	0,94 (0,06)
butte-data-set	0,62 (0,13)	0,68 (0,08)	0,67 (0,11)	0,65 (0,08)	0,65 (0,13)	0,71 (0,07)	0,61 (0,08)	0,61 (0,08)	0,64 (0,11)	0,81 (0,09)	0,86 (0,12)	0,86 (0,11)
nielsen-soft-2002	0,71 (0,16)	0,74 (0,14)	0,74 (0,13)	0,72 (0,14)	0,76 (0,11)	0,73 (0,12)	0,66 (0,15)	0,66 (0,15)	0,72 (0,20)	0,96 (0,04)	0,95 (0,07)	0,95 (0,07)
prostate-outcome	0,40 (0,39)	0,18 (0,29)	0,43 (0,41)	0,33 (0,37)	0,43 (0,33)	0,33 (0,37)	0,23 (0,25)	0,23 (0,25)	0,38 (0,32)	0,25 (0,42)	0,35 (0,47)	0,35 (0,47)
alizadeh-lymph	0,85 (0,08)	0,83 (0,12)	0,82 (0,07)	0,81 (0,06)	0,79 (0,09)	0,81 (0,12)	0,90 (0,06)	0,90 (0,06)	0,74 (0,12)	0,99 (0,03)	1,00 (0,01)	1,00 (0,01)
MLL-completo	0,89 (0,09)	0,89 (0,10)	0,87 (0,12)	0,88 (0,09)	0,89 (0,10)	0,89 (0,10)	0,89 (0,09)	0,89 (0,09)	0,72 (0,12)	0,98 (0,03)	0,99 (0,02)	0,99 (0,02)
lungcancer-ontario	0,62 (0,33)	0,63 (0,36)	0,62 (0,33)	0,59 (0,33)	0,54 (0,34)	0,59 (0,33)	0,80 (0,23)	0,80 (0,23)	0,57 (0,29)	0,65 (0,39)	0,65 (0,39)	0,65 (0,39)
DLBCLOutcome	0,50 (0,25)	0,44 (0,20)	0,50 (0,22)	0,49 (0,23)	0,56 (0,19)	0,49 (0,23)	0,51 (0,12)	0,51 (0,12)	0,44 (0,28)	0,43 (0,30)	0,43 (0,31)	0,45 (0,28)
pomeroy-cns-dataset-C	0,60 (0,30)	0,62 (0,22)	0,55 (0,24)	0,61 (0,18)	0,60 (0,30)	0,63 (0,23)	0,50 (0,12)	0,50 (0,12)	0,62 (0,15)	0,59 (0,24)	0,66 (0,19)	0,63 (0,19)
breastCancer-completo	0,60 (0,11)	0,62 (0,14)	0,56 (0,18)	0,57 (0,11)	0,58 (0,13)	0,65 (0,14)	0,62 (0,19)	0,62 (0,19)	0,50 (0,17)	0,79 (0,13)	0,78 (0,11)	0,77 (0,12)
lung-harvard	0,90 (0,09)	0,90 (0,08)	0,93 (0,07)	0,89 (0,08)	0,92 (0,06)	0,92 (0,08)	0,94 (0,06)	0,94 (0,06)	0,79 (0,09)	0,98 (0,02)	0,99 (0,01)	0,98 (0,02)
ma-2003-breast	0,55 (0,17)	0,57 (0,23)	0,50 (0,23)	0,57 (0,20)	0,49 (0,19)	0,57 (0,25)	0,49 (0,16)	0,49 (0,16)	0,48 (0,18)	0,42 (0,18)	0,38 (0,15)	0,39 (0,15)
aml-all-completo	0,75 (0,17)	0,76 (0,16)	0,81 (0,13)	0,84 (0,15)	0,82 (0,13)	0,84 (0,15)	0,75 (0,22)	0,75 (0,22)	0,66 (0,19)	0,99 (0,03)	1,00 (0,00)	1,00 (0,00)
DLBCLTumor	0,67 (0,17)	0,68 (0,15)	0,74 (0,24)	0,73 (0,26)	0,74 (0,17)	0,70 (0,19)	0,65 (0,23)	0,65 (0,23)	0,69 (0,17)	0,97 (0,06)	0,98 (0,04)	0,98 (0,04)
lung-2002-gastric	0,84 (0,17)	0,84 (0,20)	0,87 (0,11)	0,86 (0,12)	0,88 (0,15)	0,88 (0,12)	0,92 (0,11)	0,92 (0,11)	0,81 (0,16)	0,99 (0,02)	0,99 (0,02)	0,99 (0,02)
lung-Michigan	0,93 (0,16)	0,89 (0,21)	0,94 (0,16)	0,94 (0,16)	0,93 (0,15)	0,89 (0,21)	0,99 (0,02)	0,99 (0,02)	0,72 (0,26)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)
prostate-tumorVSNormal	0,86 (0,12)	0,91 (0,10)	0,85 (0,14)	0,90 (0,09)	0,83 (0,14)	0,89 (0,10)	0,79 (0,11)	0,79 (0,11)	0,68 (0,09)	0,95 (0,08)	0,95 (0,08)	0,95 (0,08)
lungCancer-completo	0,88 (0,12)	0,91 (0,12)	0,90 (0,08)	0,92 (0,09)	0,91 (0,10)	0,93 (0,08)	0,92 (0,08)	0,92 (0,08)	0,75 (0,14)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)
ovarian-61902	0,94 (0,05)	0,95 (0,05)	0,95 (0,06)	0,96 (0,05)	0,95 (0,04)	0,97 (0,04)	0,95 (0,05)	0,95 (0,05)	0,81 (0,11)	1,00 (0,01)	1,00 (0,01)	1,00 (0,01)
DLBCL-NIH-completo	0,56 (0,08)	0,61 (0,14)	0,54 (0,07)	0,64 (0,08)	0,53 (0,09)	0,62 (0,11)	0,51 (0,13)	0,50 (0,14)	0,54 (0,05)	0,64 (0,11)	0,66 (0,11)	0,66 (0,11)
BCR-ABL-completo	0,59 (0,17)	0,56 (0,17)	0,65 (0,29)	0,53 (0,12)	0,55 (0,18)	0,56 (0,18)	0,57 (0,28)	0,57 (0,28)	0,60 (0,21)	0,96 (0,04)	0,99 (0,02)	0,99 (0,02)
E2A-PBX1-completo	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	0,74 (0,14)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)
Hyperdip50-completo	0,80 (0,09)	0,77 (0,09)	0,81 (0,10)	0,83 (0,08)	0,82 (0,08)	0,81 (0,09)	0,83 (0,08)	0,83 (0,08)	0,71 (0,14)	0,99 (0,01)	0,99 (0,01)	0,99 (0,01)
T-ALL-completo	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)	1,00 (0,01)	1,00 (0,01)	0,83 (0,14)	1,00 (0,00)	1,00 (0,00)	1,00 (0,00)
Rank Medio	7,79 (7,62)	7,33 (7,71)	6,94 (7,81)	6,94 (6,34)	7,21 (6,83)	6,50 (7,19)	7,41 (6,44)	7,46 (6,50)	10,12 (8,81)	4,02 (4,60)	3,14 (4,04)	3,12 (4,00)

Tabela 5.2: Resultados do teste de Friedman para os valores de AUC para os 12 classificadores.

[illegible]

Tabela 5.3: $\log_{10}$ (Número de nós das árvores para os 12 classificadores) e desvio padrão.

Base de dados	A1-a(100)	A1-2a(100)	A1-a(70)	A1-2a(70)	A1-a(90)	A1-2a(90)	J48	J48(U)	RT	RF-128	RF-a	RF-2a
welsh-2001-GNF	0,98 (1,65)	1,03 (0,63)	0,97 (1,58)	1,03 (0,63)	0,99 (1,69)	1,03 (0,63)	1,19 (1,26)	1,19 (1,26)	1,49 (3,33)	3,50 (127,84)	5,50 (11994,41)	5,80 (24085,36)
butte-data-set	1,26 (1,90)	1,24 (1,26)	1,26 (2,15)	1,25 (1,35)	1,26 (1,93)	1,25 (1,35)	1,29 (0,97)	1,29 (0,97)	1,69 (4,37)	3,71 (54,82)	5,46 (1694,76)	5,77 (3314,13)
nielsen-soft-2002	1,06 (1,26)	1,06 (1,58)	1,06 (1,90)	1,05 (1,75)	1,06 (1,84)	1,05 (1,75)	1,03 (0,84)	1,03 (0,84)	1,40 (3,58)	3,43 (64,93)	5,07 (2017,28)	5,37 (4000,00)
prostate-outcome	0,68 (0,63)	0,60 (1,05)	0,66 (0,84)	0,62 (1,03)	0,66 (0,84)	0,62 (1,03)	0,60 (1,05)	0,60 (1,05)	0,89 (1,40)	2,95 (12,45)	4,94 (1544,84)	5,24 (3058,18)
alizadeh-lymph	1,23 (1,48)	1,20 (1,41)	1,23 (1,48)	1,21 (1,65)	1,23 (1,14)	1,20 (1,41)	1,18 (1,33)	1,18 (1,33)	1,62 (3,55)	3,65 (76,89)	5,15 (2793,06)	5,45 (5598,30)
MLL-completo	0,85 (1,33)	0,81 (1,35)	0,86 (1,14)	0,81 (1,35)	0,81 (1,35)	0,81 (1,35)	0,70 (0,00)	0,70 (0,00)	1,32 (2,39)	3,34 (41,43)	5,34 (4140,53)	5,64 (8443,43)
lung-cancer-ontario	0,79 (1,03)	0,78 (1,41)	0,82 (0,84)	0,79 (1,40)	0,82 (0,84)	0,79 (1,40)	0,70 (0,00)	0,70 (0,00)	1,22 (2,46)	3,20 (92,52)	4,55 (2026,73)	4,85 (3959,00)
DLBCLOutcome	0,98 (1,35)	0,97 (1,26)	1,00 (1,41)	0,97 (1,26)	0,97 (0,84)	0,97 (1,26)	0,91 (1,03)	0,91 (1,03)	1,32 (2,39)	3,31 (27,23)	5,06 (1094,37)	5,36 (2164,65)
pomero-y-cns-dataset-C	0,99 (1,03)	0,96 (1,14)	0,97 (0,84)	0,97 (1,26)	0,99 (1,03)	0,96 (1,14)	0,90 (1,05)	0,90 (1,05)	1,28 (1,99)	3,29 (36,12)	5,04 (1480,86)	5,34 (2938,35)
breastCancer-completo	1,12 (1,14)	1,10 (0,84)	1,12 (1,14)	1,10 (1,26)	1,11 (1,63)	1,11 (1,33)	1,06 (1,35)	1,06 (1,35)	1,47 (1,14)	3,48 (32,02)	5,76 (4838,67)	6,06 (9510,87)
lung-harvard	1,18 (1,48)	1,18 (1,75)	1,20 (1,40)	1,18 (1,63)	1,20 (1,70)	1,18 (1,63)	1,04 (0,00)	1,04 (0,00)	1,64 (5,10)	3,68 (64,91)	5,67 (7710,42)	5,97 (15527,99)
ma-2003-breast	1,14 (1,49)	1,09 (1,35)	1,13 (1,58)	1,09 (1,40)	1,14 (1,49)	1,09 (1,40)	1,11 (0,63)	1,12 (0,32)	1,47 (3,24)	3,47 (58,66)	4,65 (578,79)	4,95 (1175,47)
aml-all-completo	0,75 (0,97)	0,76 (1,03)	0,75 (0,97)	0,73 (0,84)	0,75 (0,97)	0,75 (0,97)	0,64 (0,97)	0,64 (0,97)	1,19 (2,95)	3,23 (15,09)	4,97 (1078,48)	5,27 (2265,65)
DLBCLTumor	0,85 (0,94)	0,83 (0,63)	0,86 (1,14)	0,85 (0,00)	0,86 (1,14)	0,85 (0,00)	0,79 (1,03)	0,79 (1,03)	1,18 (2,74)	3,23 (48,53)	4,97 (1595,72)	5,27 (3234,82)
lung-2002-gastric	0,97 (0,84)	0,96 (0,63)	0,97 (0,84)	0,96 (0,63)	0,98 (0,97)	0,96 (0,63)	0,92 (0,97)	0,92 (0,97)	1,50 (4,09)	3,50 (96,37)	5,22 (3627,53)	5,52 (7099,53)
lung-Michigan	0,48 (0,00)	0,48 (0,00)	0,53 (0,84)	0,51 (0,63)	0,56 (0,97)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,88 (2,32)	2,98 (23,34)	4,72 (584,94)	5,03 (1261,95)
prostate-tumorVSNormal	1,06 (1,35)	1,05 (1,14)	1,06 (1,65)	1,06 (1,65)	1,08 (1,94)	1,06 (1,65)	1,07 (2,15)	1,07 (2,15)	1,55 (3,94)	3,55 (79,28)	5,55 (6457,51)	5,85 (12661,41)
lungCancer-completo	0,75 (0,97)	0,72 (0,63)	0,78 (1,05)	0,73 (0,84)	0,76 (1,03)	0,73 (0,84)	0,82 (1,26)	0,82 (1,26)	1,26 (5,27)	3,31 (48,12)	5,30 (2767,56)	5,60 (5639,96)
ovarian-61902	0,87 (1,58)	0,85 (0,00)	0,87 (1,58)	0,85 (0,00)	0,87 (0,84)	0,85 (0,00)	1,01 (1,69)	1,01 (1,69)	1,59 (4,16)	3,60 (70,97)	5,67 (3470,19)	5,98 (6763,88)
DLBCL-NHL-completo	1,58 (3,02)	1,53 (1,69)	1,57 (2,49)	1,53 (1,90)	1,58 (3,02)	1,53 (2,35)	1,50 (3,29)	1,55 (6,45)	2,02 (6,33)	4,02 (131,46)	5,79 (4520,41)	6,09 (9349,92)
BCR-ABL-completo	0,93 (1,58)	0,91 (1,03)	0,98 (1,90)	0,91 (1,03)	0,94 (1,75)	0,91 (1,03)	0,95 (0,00)	0,95 (0,00)	1,51 (5,32)	3,52 (134,91)	5,52 (8970,52)	5,82 (18209,36)
E2A-PBX1-completo	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	1,45 (6,34)	3,53 (102,94)	5,52 (6449,84)	5,82 (12681,04)
Hyperdip50-completo	1,20 (1,41)	1,18 (1,14)	1,23 (1,14)	1,20 (1,03)	1,21 (1,03)	1,19 (0,97)	1,23 (2,49)	1,23 (2,49)	1,81 (9,20)	3,81 (132,42)	5,81 (6894,66)	6,11 (13835,23)
T-ALL-completo	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	0,48 (0,00)	1,54 (9,11)	3,55 (67,92)	5,55 (6870,10)	5,86 (11593,31)
Rank Médio	5,54 (4,52)	3,44 (4,04)	6,10 (5,16)	3,75 (4,23)	6,06 (5,27)	3,73 (4,23)	3,58 (4,40)	3,79 (4,44)	9,00 (8,71)	10,00 (10,00)	11,00 (11,00)	12,00 (12,00)

Tabela 5.4: Resultados do teste de Friedman para os números de nós dos 9 classificadores.

Classificador	A1-a(100)	A1-2a(100)	A1-a(70)	A1-2a(70)	A1-a(90)	A1-2a(90)	J48	J48(U)	RT
A1-a(100)	○	▼	△	▼	△	▼	▼	▼	▲
A1-2a(100)		○	▲	△	▲	△	△	△	▲
A1-a(70)			○	▼	▽	▼	▼	▼	▲
A1-2a(70)				○	▲	▽	▽	△	▲
A1-a(90)					○	▼	▼	▼	▲
A1-2a(90)						○	▽	△	▲
J48							○	△	▲
J48(U)								○	▲
RT									○

Tabela 5.5: Valores das métricas de comparação entre árvores geradas pelo Algoritmo 1 (A1-2a(90)) e pelo algoritmo J48

Base de dados	Métrica J	Métrica JN	Métrica JNP	Métrica P
welsh-2001-GNF	0,01	0,00	0,00	0,00
butte-data-set	0,01	0,02	0,01	0,00
nielsen-soft-2002	0,04	0,00	0,00	0,00
prostate-outcome	0,18	0,20	0,02	0,03
alizadeh-lymph	0,08	0,05	0,00	0,00
MLL-completo	0,57	0,60	0,18	0,09
lungcancer-ontario	0,39	0,43	0,28	0,30
DLBCLOutcome	0,06	0,08	0,04	0,06
pomeroy-cns-dataset-C	0,09	0,14	0,09	0,07
breastCancer-completo	0,07	0,06	0,03	0,02
lung-harvard	0,15	0,24	0,06	0,07
ma-2003-breast	0,05	0,03	0,00	0,00
aml-all-completo	0,10	0,15	0,13	0,13
DLBCLTumor	0,10	0,20	0,16	0,15
leung-2002-gastric	0,14	0,03	0,00	0,00
lung-Michigan	0,00	0,00	0,00	0,00
prostate-tumorVSNormal	0,15	0,28	0,08	0,09
lungCancer-completo	0,16	0,22	0,03	0,02
ovarian-61902	0,13	0,27	0,24	0,21
DLBCL-NIH-completo	0,03	0,03	0,03	0,02
BCR-ABL-completo	0,01	0,03	0,01	0,01
E2A-PBX1-completo	0,00	0,00	0,00	0,00
Hyperdip50-completo	0,04	0,02	0,02	0,02
T-ALL-completo	1,00	1,00	0,00	0,00
Média	0,15	0,17	0,06	0,05
Mediana	0,09	0,07	0,03	0,02

Tabela 5.6: Valores das métricas de comparação entre árvores geradas pelo Algoritmo 1 ($A1-2a(90)$) e pela Random Tree

Base de dados	Métrica J	Métrica JN	Métrica JNP	Métrica P
welsh-2001-GNF	0,01	0,00	0,00	0,00
butte-data-set	0,00	0,00	0,00	0,00
nielsen-soft-2002	0,01	0,00	0,00	0,00
prostate-outcome	0,00	0,00	0,00	0,00
alizadeh-lymph	0,01	0,00	0,00	0,00
MLL-completo	0,00	0,00	0,00	0,00
lungcancer-ontario	0,00	0,00	0,00	0,00
DLBCLOutcome	0,00	0,00	0,00	0,00
pomeroiy-cns-dataset-C	0,01	0,01	0,00	0,00
breastCancer-completo	0,01	0,00	0,00	0,00
lung-harvard	0,00	0,00	0,00	0,00
ma-2003-breast	0,02	0,00	0,00	0,00
aml-all-completo	0,00	0,00	0,00	0,00
DLBCLTumor	0,00	0,00	0,00	0,00
leung-2002-gastric	0,00	0,00	0,00	0,00
lung-Michigan	0,00	0,00	0,00	0,00
prostate-tumorVSNormal	0,00	0,00	0,00	0,00
lungCancer-completo	0,00	0,00	0,00	0,00
ovarian-61902	0,00	0,00	0,00	0,00
DLBCL-NIH-completo	0,00	0,00	0,00	0,00
BCR-ABL-completo	0,00	0,00	0,00	0,00
E2A-PBX1-completo	0,00	0,00	0,00	0,00
Hyperdip50-completo	0,00	0,00	0,00	0,00
T-ALL-completo	0,00	0,00	0,00	0,00
Média	0,00	0,00	0,00	0,00
Mediana	0,00	0,00	0,00	0,00

Tabela 5.7: Valores dos coeficientes de regressão.

Métricas comparadas	R^2
J e JN	0,94
J e JNP	0,07
J e M_4	0,04
JN e JNP	0,16
JN e P	0,11
JNP e P	0,93

Conclusões

Após a execução dos dois experimentos preliminares explicados anteriormente foi gerada a árvore a partir de uma *Random Forest*. Tal árvore foi gerada usando *Random Forests* com a e $2a$ árvores, a frequência considerando o erro *out-of-bag* estimado como métrica de escolha dos atributos que iriam compor a árvore final e uma porcentagem desses atributos a serem considerados (100%, 90% e 70%). Foram analisadas 24 bases de expressão gênica (descritas na Seção A.2). No geral, o algoritmo proposto apresentou um desempenho igual ou melhor do que o classificador J48 e *Random Tree* e um desempenho inferior ao da *Random Forest*. A árvore também apresentou, no geral, números de nós similares ao J48 e menores do que a *Random Tree*.

Quando comparadas as estruturas das árvores criadas pelo Algoritmo 1, J48 e *Random Tree* pôde-se concluir que as árvores obtidas pelo Algoritmo 1 são mais parecidas com as obtidas pelo J48, porém essa similaridade não é alta. Essa diferença nas estruturas das árvores, ou seja, nos genes utilizados por cada uma para classificar novos exemplos, pode ser muito interessante, uma vez que as informações obtidas por meio de cada uma são diferentes. Assim, o algoritmo aqui proposto pode fornecer informações diferentes a respeito das bases

de expressão gênica com relação aos algoritmos existentes até o momento, como por exemplo J48 e *Random Tree*.

Assim, analisando os resultados obtidos, pode-se notar que a metodologia proposta neste estudo tem um bom potencial, uma vez que apresenta desempenho melhor do que alguns classificadores já conhecidos em muitas bases de dados testadas e número de nós menores, ou seja, árvores menores e mais simples de serem interpretadas por humanos. Além disso, as árvores criadas apresentaram estruturas (nós) diferentes das árvores geradas pelos demais classificadores e, assim, elas podem fornecer informações diferentes e até novas a respeito de um determinado problema biológico, ampliando assim o conhecimento humano.

6.1 Principais resultados e publicações

O presente trabalho resultou em 2 publicações em dois congressos internacionais.

O primeiro artigo (Oshiro, Perez & Baranauskas 2012), intitulado *How Many Trees in a Random Forest?*, foi aceito no *8th International Conference on Machine Learning and Data Mining* (MLDM 2012) e publicado em *Lecture Notes in Computer Science*. Este artigo é referente ao primeiro experimento realizado durante o desenvolvimento do projeto de mestrado, descrito na Seção B.2. DOI 10.1007/978-3-642-31537-4_13.

O segundo artigo (Oshiro & Baranauskas 2012), intitulado *Root Attribute Behavior within a Random Forest*, foi aceito no *Intelligent Data Engineering and Automated Learning* (IDEAL 2012) e publicado em *Lecture Notes in Computer Science*. Este artigo tem como tema o segundo experimento, descrito na Seção B.3. DOI 10.1007/978-3-642-32639-4_87.

6.2 Trabalhos Futuros

Os resultados obtidos neste projeto de mestrado são promissores, uma vez que a árvore final obtida apresentou uma estabilidade maior do que a *Random Tree* (como esperado) e na maioria dos casos, maior do que o J48 e apresentou um *rank* médio melhor do que o J48 e a *Random Tree* para os valores de AUC obtidos e os números de nós das árvores criadas. Sendo assim, trabalhos futuros poderão melhorar o desempenho e a estabilidade do algoritmo proposto.

Outros métodos para a criação da árvore final poderão ser desenvolvidos e seus resultados,

poderão ser comparados aos obtidos neste trabalho.

Referências

- (2010). Cancer program data sets. Broad Institute. Citado na página 73.
- (2010). Dataset repository in arff (weka). BioInformatics Group Seville. Citado na página 73.
- (2010). Datasets. <http://cilab.ujn.edu.cn/datasets.htm>. Cilab. Citado na página 73.
- Acuna, E. & Rojas, A. (2001). Bagging classifiers based on kernel density estimators. In *Proceedings of the International Conference on New Trends in Computational Statistics with Biomedical Applications*, pp. 343–350. Citado na página 16.
- Alberts, B. (1997). *Biologia molecular da célula* (3 ed.). Artes Médicas. Citado na página 29.
- Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. & Walter, P. (2010). *Biologia Molecular da Célula* (5 ed.). Porto Alegre. Citado nas páginas vi, 20, 21, 22, 23, 24, 25, 26, 27, 28, and 29.
- Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O. & Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* 403(6769), 503–511. Citado na página 73.

- Armstrong, S. A., Staunton, J. E., Silverman, L. B., Pieters, R., den Boer, M. L., Minden, M. D., Sallan, S. E., Lander, E. S., Golub, T. R. & Korsmeyer, S. J. (2002). MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat Genet* 30, 41–47. Citado na página 73.
- Aslan, O., Yildiz, O. T. & Alpaydin, E. (2009). Calculating the VC-dimension of decision trees. In *International Symposium on Computer and Information Sciences 09*, pp. 193–198. Citado na página 77.
- Aziz, A. & Ahmed, N. (2011). Construction of single classifier from multiple interim classification trees. *International Journal of Computer Science and Network Security* 11(5), 172–178. Citado na página 36.
- Baranauskas, J. A. (2001). Extração automática de conhecimento utilizando múltiplos indutores. Technical report, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Brasil. Citado na página 12.
- Bauer, E. & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning* 36(1-2), 105–139. Citado na página 16.
- Beer, D. G., Kardia, S. L., Huang, C.-C., Giordano, T. J., Levin, A. M., Misek, D. E., Lin, L., Chen, G., Gharib, T. G., Thomas, D. G., Lizyness, M. L., Kuick, R., Hayasaka, S., Taylor, J. M., Iannettoni, M. D., Orringer, M. B. & Hanash, S. (2002). Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat Med* 8(8), 816–824. Citado na página 74.
- Benjamini, Y. & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B* 57, 289–300. Citado na página 41.
- Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T. R., Sugarbaker, D. J. & Meyerson, M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences of the*

- United States of America* 98(24), 13790–13795. Citado na página 74.
- Breiman, L. (1996). Bagging predictors. *Machine Learning* 24(2), 123–140. Citado nas páginas 1, 14, 15, and 16.
- Breiman, L. (2001). Random forests. *Machine Learning* 45(1), 5–32. Citado nas páginas 1, 14, 15, 16, 17, and 18.
- Breiman, L. (2004). Wald lecture ii, looking inside the black box. <http://www.stat.berkeley.edu/users/breiman>. Citado na página 18.
- Breiman, L. & Cutler, A. (2004). Random forests: Classification/clustering. <http://www.stat.berkeley.edu/users/breiman/RandomForests>. Citado na página 18.
- Brenner, S., Johnson, M., Bridgham, J., Golda, G., Lloyd, D. H., Johnson, D., Luo, S., McCurdy, S., Foy, M., Ewan, M., Roth, R., George, D., Eletr, S., Albrecht, G., Vermaas, E., Williams, S. R., Moon, K., Burcham, T., Pallas, M., DuBridge, R. B., Kirchner, J., Fearon, K., Mao, J. & Corcoran, K. (2000). Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nature biotechnology* 18(6), 630–634. Citado na página 30.
- Bruce, R. (2001). A bayesian approach to semi-supervised learning. *North*, 57–64. Citado na página 6.
- Butte, A. J., Tamayo, P., Slonim, D., Golub, T. R. & Kohane, I. S. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences of the United States of America* 97(22), 12182–12186. Citado na página 73.
- Cheung, V. G., Morley, M., Aguilar, F., Massimi, A., Kucherlapati, R. & Childs, G. (1999). Making and reading microarrays. *Nature genetics* 21(1 Suppl), 15–19. Citado na página 30.
- Díaz-Uriarte, R. & de Andrés, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics* 7, 3. Citado nas páginas 2, 3, 34, and 35.
- de Souto, M. C. P., Lorena, A., Delbem, A. & de Carvalho, A. (2003). Técnicas de Aprendi-

- zado de Máquina para Problemas de Biologia Molecular. In *III Jornada de Inteligência Artificial*. Citado na página 29.
- Demšar, J. (2006). Statistical comparison of classifiers over multiple data sets. *Journal of Machine Learning Research* 7(1), 1–30. Citado nas páginas 40 and 41.
- Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science* 1857, 1–15. Citado nas páginas 1, 13, and 16.
- Dubath, P., Rimoldini, L., Süveges, M., Blomme, J., López, M., Sarro, L. M., De Ridder, J., Cuypers, J., Guy, L., Lecoeur, I., Nienartowicz, K., Jan, A., Beck, M., Mowlavi, N., De Cat, P., Lebzelter, T. & Eyer, L. (2011). Random forest automated supervised classification of hipparcos periodic variable stars. *Monthly Notices of the Royal Astronomical Society* 414(3), 2602–2617. Citado na página 17.
- Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics* 7(1), 1–26. Citado na página 15.
- Frank, A. & Asuncion, A. (2010). UCI machine learning repository. <http://archive.ics.uci.edu/ml>. Citado na página 73.
- Freund, Y. & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, Lake Tahoe, California, pp. 148–156. Citado nas páginas 1, 14, and 16.
- Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics* 11(1), 86–92. Citado nas páginas 41, 76, and 88.
- Gamberger, D., Lavrac, N., Zelezny, F. & Tolar, J. (2004). Induction of comprehensible models for gene expression datasets by subgroup discovery methodology. *Journal of Biomedical Informatics* 37, 269–284. Citado nas páginas 3 and 84.
- Goldstein, B., Hubbard, A., Cutler, A. & Barcellos, L. (2010). An application of random forests to a genome-wide association dataset: Methodological considerations and new findings. *BMC Genetics* 11(1), 49. Citado nas páginas 34, 35, and 39.
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P.,

- Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. & Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286(5439), 531–537. Citado na página 74.
- Gordon, G. J., Jensen, R. V., Li Hsiao, L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., Richards, W. G., Sugarbaker, D. J. & Bueno, R. (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research* 62(17), 4963. Citado na página 74.
- Guindalini, C. & Tufik, S. (2007). Use of microarrays in the search of gene expression patterns: application to the study of complex phenotypes. *Revista brasileira de psiquiatria Sao Paulo Brazil 1999* 29(4), 370–374. Citado na página 30.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2009). The weka data mining software: an update. *Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining Explor. Newsl.* 11(1), 10–18. Citado nas páginas 40, 85, and 88.
- Hand, D. & Till, R. (2001). **A simple generalisation of the area under the ROC curve for multiple class classification problems.** *Machine Learning* 45(2), 171–186. Citado na página 8.
- Huang, J. & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on* 17(3), 299–310. Citado na página 9.
- Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles* 37, 547–579. Citado na página 41.
- Klassen, M., Cummings, M. & Saldaña, G. (2008). Investigation of random forest performance with cancer microarray data. In T. Philip (Ed.), *Computers and Their Applications*, pp. 64–69. ISCA. Citado na página 2.
- Klug, W., Cummings, M., Palladino, M. & Spencer, C. (2010). *Conceitos de Genética* (9 ed.). ArtMed Editora. Citado nas páginas vi, 20, 21, 22, 23, 25, 27, 28, 31, and 33.

- Kubat, M., Bratko, I. & Michalski, R. (1998). A review of machine learning methods. Citado na página 6.
- Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience. Citado na página 1.
- Lee, J., Park, M. & Song, S. (2005). An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics Data Analysis* 48(4), 869–885. Citado na página 34.
- Leung, S. Y., Chen, X., Chu, K. M., Yuen, S. T., Mathy, J., Ji, J., Chan, A. S. Y., Li, R., Law, S., Troyanskaya, O. G., Tu, I.-P., Wong, J., So, S., Botstein, D. & Brown, P. O. (2002). Phospholipase a2 group iia expression in gastric adenocarcinoma is associated with prolonged survival and less frequent metastasis. *Proceedings of the National Academy of Sciences* 99(25), 16203. Citado na página 74.
- Lewis, R. (2001). *Human genetics: concepts and applications* (4 ed.). McGraw-Hill higher education. McGraw-Hill. Citado na página 22.
- Liaw, A. & Wiener, M. (2002). Classification and regression by randomforest. *R News* 2(3), 18–22. Citado nas páginas 1, 14, and 37.
- Ling, C., Huang, J. & Zhang, H. (2003). Auc: A better measure than accuracy in comparing learning algorithms. In Y. Xiang & B. Chaib-draa (Eds.), *Advances in Artificial Intelligence*, Volume 2671 of *Lecture Notes in Computer Science*, pp. 991–991. Springer Berlin / Heidelberg. Citado nas páginas 10 and 39.
- Liu, E. T. Citado na página 30.
- Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Wang, C., Kobayashi, M., Norton, H. & Brown, E. L. (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology* 14(13), 1675–1680. Citado na página 30.
- Lopes, L. (2007). Aprendizagem de máquina baseada na combinação de classificadores em bases de dados da área de saúde. Master’s thesis, Pontifícia Universidade Católica do Paraná. Citado na página 16.

- Ma, X.-J., Salunga, R., Tuggle, J. T., Gaudet, J., Enright, E., McQuary, P., Payette, T., Pistone, M., Stecker, K., Zhang, B. M., Zhou, Y.-X., Varnholt, H., Smith, B., Gadd, M., Chatfield, E., Kessler, J., Baer, T. M., Erlander, M. G. & Sgroi, D. C. (2003). Gene expression profiles of human breast cancer progression. *Proceedings of the National Academy of Sciences* 100(10), 5974–5979. Citado na página 74.
- Ma, Y., Guo, L. & Cukic, B. (2007). Statistical framework for the prediction of fault-proneness. In *Advances in machine learning applications in software engineering*. Idea Group. Citado na página 18.
- Michalski, R. S. (1983). A theory and methodology of inductive learning. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning : An artificial intelligence approach*, Volume 1, pp. 83–134. Morgan Kaufmann. Citado na página 6.
- Mitchell, T. (1997). *Machine Learning (Mcgraw-Hill International Edit)*. McGraw-Hill Education (ISE Editions). Citado nas páginas 6 and 76.
- Monard, M. C. & Baranauskas, J. A. (2003). *Indução de Regras e Árvores de Decisão*, Chapter 5, pp. 115–140. Manole. Citado na página 6.
- Nanni, L., Brahnam, S. & Lumini, A. (2012). Combining multiple approaches for gene microarray classification. *Bioinformatics* 28(8), 1151–1157. Citado na página 2.
- Netto, O. P., Nozawa, S. R., Mitrowsky, R. A. R., Macedo, A. A. & Baranauskas, J. A. (2010). Applying decision trees to gene expression data from dna microarrays: A leukemia case study. In *XXX Congresso da Sociedade Brasileira de Computação*, Belo Horizonte, MG, pp. 10p. Citado nas páginas 72 and 84.
- Nielsen, T., West, R., Linn, S., Alter, O., Knowling, M., Oconnell, J., Zhu, S., Fero, M., Sherlock, G. & Pollack, J. (2002). Molecular characterisation of soft tissue tumours: a gene expression study. *The Lancet* 359(9314), 1301–1307. Citado na página 73.
- Oh, I.-S., Lee, J.-S. & Moon, B.-R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 1424–1437. Citado na página 39.
- Oshiro, T. M. & Baranauskas, J. A. (2012). Root attribute behavior within a random forest. In *Intelligent Data Engineering and Automated Learning - IDEAL 2012*, Volume

- 7435 of *Lecture Notes in Computer Science*, pp. 733–742. Springer Berlin Heidelberg. Citado nas páginas 39, 40, 45, 58, and 71.
- Oshiro, T. M., Perez, P. S. & Baranauskas, J. A. (2012). How many trees in a random forest? In P. Perner (Ed.), *Machine Learning and Data Mining in Pattern Recognition*, Volume 7376 of *Lecture Notes in Computer Science*, pp. 154–168. Springer Berlin Heidelberg. Citado nas páginas 36, 37, 39, 45, 58, and 71.
- Pang, H., Lin, A., Holford, M., Enerson, B. E., Lu, B., Lawton, M. P., Floyd, E. & Zhao, H. (2006). Pathway analysis using random forests classification and regression. *Bioinformatics* 22, 2028–2036. Citado na página 3.
- Perez, P. S. & Baranauskas, J. A. (2011). Analysis of decision tree pruning using windowing in medical datasets with different class distributions. In *Proceedings of the Workshop on Knowledge Discovery in Health Care and Medicine of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD KDHCM)*, Athens, Greece, pp. 28–39. Citado na página 77.
- Perner, P. (2011). How to interpret decision trees? In *Proceedings of the 11th international conference on Advances in data mining: applications and theoretical aspects*, ICDM'11, Berlin, Heidelberg, pp. 40–55. Springer-Verlag. Citado nas páginas 41 and 43.
- Petricoin, E. F., Ardekani, A. M., Hitt, B. A., Levine, P. J., Fusaro, V. A., Steinberg, S. M., Mills, G. B., Simone, C., Fishman, D. A., Kohn, E. C. & Liotta, L. A. (2002). Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet* 359(9306), 572–577. Citado na página 74.
- Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., Kim, J. Y. H., Goumnerova, L. C., Black, P. M., Lau, C., Allen, J. C., Zagzag, D., Olson, J. M., Curran, T., Wetmore, C., Biegel, J. A., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D. N., Mesirov, J. P., Lander, E. S. & Golub, T. R. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature* 415(6870), 436–442. Citado na página 74.
- Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., Gascoyne, R. D., Muller-Hermelink, H. K., Smeland, E. B., Giltane, J. M., Hurt, E. M.,

- Zhao, H., Averett, L., Yang, L., Wilson, W. H., Jaffe, E. S., Simon, R., Klausner, R. D., Powell, J., Duffey, P. L., Longo, D. L., Greiner, T. C. & It Et Al (2002). The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *New England Journal of Medicine* 346(25), 1937–1947. Citado na página 75.
- Saeys, Y., Inza, I. n. & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics* 23, 2507–2517. Citado na página 39.
- Sharp, F. R., Xu, H., Lit, L., Walker, W., Apperson, M., Gilbert, D. L., Glauser, T. A., Wong, B., Hershey, A., Liu, D.-Z. Z., Pinter, J., Zhan, X., Liu, X. & Ran, R. (2006). The future of genomic profiling of neurological diseases using blood. *Archives of neurology* 63(11), 1529–1536. Citado na página 30.
- Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C., Gaasenbeek, M., Angelo, M., Reich, M., Pinkus, G. S., Ray, T. S., Koval, M. A., Last, K. W., Norton, A., Lister, T. A., Mesirov, J., Neuberg, D. S., Lander, E. S., Aster, J. C. & Golub, T. R. (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine* 8(1), 68–74. Citado na página 74.
- Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D’Amico, A. V. & Richie, J. P. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell* 1(2), 203–209. Citado nas páginas 73 and 74.
- Sirikulviriy, N. & Sinthupinyo, S. (2011). Integration of rules from a random forest. *International Conference on Information and Electronics Engineering* 6, 194–198. Citado na página 1.
- Tan, P.-N., Steinbach, M. & Kumar, V. (2005). *Introduction to Data Mining* (1 ed.). Addison Wesley. Não citado no texto.
- van ’t Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R. & Friend, S. H. (2002).

- Nature* 415(6871), 530–536. Citado na página 74.
- Vapnik, V., Levin, E. & Cun, Y. L. (1994). Measuring the vc-dimension of a learning machine. *Neural Computation* 6, 851–876. Citado na página 76.
- Velculescu, V. E., Zhang, L., Vogelstein, B. & Kinzler, K. W. (1995). Serial analysis of gene expression. *Science* 270, 484–487. Citado na página 29.
- Wang, G., Hao, J., Ma, J. & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications* 38, 223–230. Citado na página 14.
- Watson, J. D. & Crick, F. H. C. (1953). Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature* 171(4356), 737–738. Citado na página 20.
- Weiss, S. M. & Kulikowski, C. A. (1991). *Computer systems that learn : classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. Morgan Kaufmann Publishers. Citado na página 5.
- Welsh, J. B., Sapinoso, L. M., Su, A. I., Kern, S. G., Wang-Rodriguez, J., Moskaluk, C. A., Frierson, H. F. & Hampton, G. M. (2001). Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Research* 61(16), 5974–8. Citado na página 73.
- Wigle, D. A., Jurisica, I., Radulovich, N., Pintilie, M., Rossant, J., Liu, N., Lu, C., Woodgett, J., Seiden, I., Johnston, M., Keshavjee, S., Darling, G., Winton, T., Breitkreutz, B.-J., Jorgenson, P., Tyers, M., Shepherd, F. A. & Tsao, M. S. (2002). Molecular profiling of non-small cell lung cancer and correlation with disease-free survival. *Cancer Research* 62(11), 3005. Citado na página 73.
- Witten, I. H. & Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Volume 1. Morgan Kaufmann. Citado na página 1.
- Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K. & Zhao, H. (2003). Comparison of statistical methods for classification of ovarian cancer using a proteomics dataset. *Bioinformatics* 19(13), 1636–1643. Citado

na página 34.

Yamamoto, M., Wakatsuki, T., Hada, A. & Ryo, A. (2001). Use of serial analysis of gene expression (SAGE) technology. *Journal of immunological methods* 250(1-2), 45–66.

Citado nas páginas 29 and 30.

Yeoh, E. J., Ross, M. E., Shurtleff, S. A., Williams, K. W., Patel, D., Mahfouz, R., Behm, F. G., Raimondi, S. C., Relling, M. V., Patel, A. & Cheng (2002). Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell* 1(2), 133–143. Citado na página 75.

Zhao, Y. & Zhang, Y. (2008). Comparison of decision tree methods for finding active objects. *Advances in Space Research* 41, 1955–1959. Citado na página 17.

Bases de Dados

Neste apêndice são descritas resumidamente as bases utilizadas nos experimentos preliminares (Oshiro, Perez & Baranauskas 2012) e (Oshiro & Baranauskas 2012) descritos no Apêndice B e na construção da árvore final. As bases referentes aos experimentos estão no Apêndice A.1 e as referentes a construção da árvore final estão no Apêndice A.2.

A.1 Bases de dados usadas para os experimentos realizados

Todas as bases de dados usadas representam bases médicas reais ou bases de expressão gênica e nenhuma possui valor ausente para o atributo classe. Na Tabela A.1 é mostrado um resumo das bases de dados e as métricas de densidade correspondentes, definidas na Seção B.1. As bases estão ordenadas de acordo com a métrica D_2 (vide Apêndice B.1), obtendo assim 8 bases de baixas densidades e 21 de altas densidades. No restante desta seção, uma breve descrição de cada base é fornecida.

Breast Cancer, Lung Cancer, CNS(Central Nervous System Tumour Outcome),Lymphoma,

Tabela A.1: *Resumo das bases de dados utilizadas neste experimento, onde n indica o número de exemplos; c representa o número de classes; a , $a_{\#}$ e a_a indica o número total de atributos, o número de atributos numéricos e o número de atributos nominais, respectivamente; MISS representa a porcentagem de atributos com valores ausentes, sem considerar o atributo classe; as últimas 3 colunas são as métricas de densidade D_1 , D_2 , D_3 de cada base, respectivamente. Bases estão ordenadas por D_2 em ordem crescente.*

Base de dados	n	c	$a(a_{\#}, a_a)$	MISS	D_1	D_2	D_3
GCM	190	14	16063 (16063, 0)	0,00%	0,54	0,27	0,26
Lymphoma	96	9	4026 (4026, 0)	5,09%	0,55	0,28	0,27
CNS	60	2	7129 (7129, 0)	0,00%	0,46	0,38	0,34
Leukemia	72	2	7129 (7129, 0)	0,00%	0,48	0,40	0,36
Leukemia nom,	72	2	7129 (7129, 0)	0,00%	0,48	0,40	0,36
Ovarian 61902	253	2	15154 (15154, 0)	0,00%	0,57	0,50	0,46
Lung Cancer	32	3	56 (0, 56)	0,28%	0,86	0,59	0,52
C. Arrhythmia	452	16	279 (206, 73)	0,32%	1,08	0,59	0,58
Dermatology	366	6	34 (1, 33)	0,06%	1,67	1,17	1,12
HD Switz,	123	5	13 (6, 7)	17,07%	1,88	1,25	1,18
Lymphography	148	4	18 (3, 15)	0,00%	1,73	1,25	1,17
Hepatitis	155	2	19 (6, 13)	5,67%	1,71	1,48	1,34
HD Hungarian	294	5	13 (6, 7)	20,46%	2,21	1,59	1,52
HD Cleveland	303	5	13 (6, 7)	0,18%	2,22	1,60	1,53
P. Patient	90	3	8 (0, 8)	0,42%	2,16	1,63	1,50
WDBC	569	2	30 (30, 0)	0,00%	1,86	1,66	1,54
Splice Junction	3190	3	60 (0, 60)	0,00%	1,97	1,70	1,63
Heart Statlog	270	2	13 (13, 0)	0,00%	2,18	1,91	1,75
Allhyper	3772	5	29 (7, 22)	5,54%	2,44	1,97	1,91
Allhypo	3772	4	29 (7, 22)	5,54%	2,44	2,03	1,97
Sick	3772	2	29 (7, 22)	5,54%	2,44	2,24	2,12
Breast Cancer	286	2	9 (0, 9)	0,35%	2,57	2,26	2,07
Hypothyroid	3163	2	25 (7, 18)	6,74%	2,50	2,29	2,16
ANN Thyroid	7200	3	21 (6, 15)	0,00%	2,92	2,56	2,46
WBC	699	2	9 (9, 0)	0,25%	2,98	2,66	2,48
C. Method	1473	3	9 (2, 7)	0,00%	3,32	2,82	2,69
Pima Diabetes	768	2	8 (8, 0)	0,00%	3,19	2,86	2,67
Liver Disorders	345	2	6 (6, 0)	0,00%	3,26	2,87	2,65
H. Survival	306	2	3 (2, 1)	0,00%	5,21	4,58	4,21

GCM(Global Cancer Map), *Ovarian 61902*, *Leukemia*, *Leukemia nom.*, *WBC* (Wisconsin Breast Cancer), *WDBC* (Wisconsin Diagnostic Breast Cancer), *Lymphography* e *H. Survival* (*H.* de *Haberman's* são todas relacionadas a cancer e seus atributos consistem de dados clínicos, laboratoriais e expressão gênica. *Leukemia* e *Leukemia nom.* representam a mesma base, mas a segunda tem seus atributos discretizados (Netto et al. 2010). *C. Arrhythmia* (*C.* de *Cardiac*), *Heart Statlog*, *HD Cleveland*, *HD Hungarian* e *HD Switz.* (*Switz.* de *Switzerland*) estão relacionados a doenças cardíacas e seus atributos representam dados clínicos e laboratoriais. *Allhyper*, *Allhypo*, *ANN Thyroid*, *Hypothyroid* e *Sick* são uma série de bases relacionadas a condições da tireóide. *Hepatitis* e *Liver Disorders* estão relacionadas com doenças do fígado, enquanto que *C. Method* (*C.* de *Contraceptive*), *Dermatology*, *Pima Diabetes* (Pima Indians Diabetes) e *P. Patient* (*P.* de *Postoperative*) são outras bases

relacionadas a condições humanas. *Splice Junction* está relacionado a tarefa de previsão de limites entre exons e introns. As bases foram obtidas no Repositório UCI (Frank & Asuncion 2010), exceto *CNS*, *Lymphoma*, *GCM* e *ECML* que foram obtidas a partir de (Dat 2010); *Ovarian 61902* foi obtida em (Dat 2010); *Leukemia* e *Leukemia nom.* foram obtidas em (Can 2010).

A.2 Bases de dados usadas para o Algoritmo 1

Foram utilizadas 24 bases de expressão gênica por *microarray*, descritas resumidamente na Tabela A.2, cuja legenda é a mesma da Tabela A.1. A seguir, uma breve descrição de cada base utilizada.

- welsh-2001-GNF está relacionada com a classificação de amostras de tecido da próstata (Welsh et al. 2001);
- butte-data-set está relacionada com a diferenciação entre diversos tipos de câncer: mama, ovário, leucemia, entre outros (Butte et al. 2000);
- nielsen-soft-2002 está relacionada com a diferenciação entre os tipos de câncer: sarcoma sinovial, tumor estromal gastrointestinal, leiomiossarcoma, lipossarcoma, histiocitoma fibroso maligno e Schwannoma (Nielsen et al. 2002);
- prostate-outcome está relacionada com a classificação do resultado clínico em pacientes com tumor, os quais foram avaliados com relação à recorrência da doença após a cirurgia em um período de quatro anos (Singh et al. 2002);
- alizadeh-lymph está relacionada com a classificação de pacientes de acordo com a variação molecular do linfoma de células B (Alizadeh et al. 2000);
- MLL-completo está relacionada com a diferenciação de pacientes com leucemia mieloide aguda, leucemia linfocítica aguda e leucemia de linhagem misturada (MLL) (Armstrong et al. 2002);
- lungcancer-ontario está relacionada com a classificação de pacientes que tiveram tumor em: pacientes que apresentaram metástase local ou distante, ou como livre da doença (Wigle et al. 2002);

- DLBCLOutcome está relacionada com a classificação de pacientes (curados ou fatais) com linfoma de células B quanto ao resultado clínico da doença (Shipp et al. 2002);
- pomeroys-cns-dataset-C está relacionada com a classificação de pacientes com meduloblastoma em pacientes que sobreviveram ao tratamento e pacientes que não sobreviveram (Pomeroy et al. 2002);
- breastCancer-completo está relacionada com a classificação de paciente com câncer de mama (van 't Veer et al. 2002);
- lung-harvard está relacionada com a diferenciação de pacientes entre os diferentes tipos de tumores e também pacientes normais (Bhattacharjee et al. 2001);
- ma-2003-breast está relacionada com a classificação de pacientes com: carcinoma ductal *in-situ*, carcinoma ductal invasivo e hiperplasia ductal atípica (Ma et al. 2003);
- aml-all-completo está relacionada com a classificação de pacientes com: leucemia mieloide aguda (AML) ou leucemia linfoblástica aguda (ALL) (Golub et al. 1999);
- DLBCLTumor está relacionada com a classificação da morfologia das amostras em linfoma de células B e linfoma folicular (Shipp et al. 2002);
- leung-2002-gastric está relacionada com a classificação de pacientes em: tumor gástrico primário, tumor com metástase e mucosa normal (Leung et al. 2002);
- lung-Michigan está relacionada com a classificação de pacientes em tendo adenocarcinoma primário de pulmão ou não tendo (Beer et al. 2002);
- prostate-tumorVSNormal está relacionada com a diferenciação entre amostra normal e amostra com tumor (Singh et al. 2002);
- lungCancer-completo está relacionada com a classificação de câncer de pulmão entre mesotelioma pleural maligno e adenocarcinoma (Gordon et al. 2002);
- ovarian-61902 está relacionada com a distinção entre pacientes com câncer e pacientes normais (Petricoin et al. 2002);

- DLBCL-NIH-completo está relacionada com a classificação de pacientes com linfoma de células B em: óbito ou sobrevivência (Rosenwald et al. 2002);
- BCR-ABL-completo está relacionada com a classificação de subtipos de leucemia linfoblástica pediátrica (Yeoh et al. 2002);
- E2A-PBX1-completo, Hyperdip50-completo e T-ALL-completo estão relacionados com a classificação de subtipos de leucemia linfoblástica pediátrica (Yeoh et al. 2002).

Tabela A.2: *Resumo das bases de dados utilizadas neste experimento.*

Base de dados	n	c	$a(a_{\#}, a_a)$	MISS	D_1	D_2	D_3
welsh-2001-GNF	55	14	12626 (12626,0)	0,00%	0,42	0,14	0,14
butte-data-set	68	9	7245 (7245,0)	0,00%	0,47	0,23	0,22
nielsen-soft-2002	46	6	5520 (5520,0)	0,00%	0,44	0,24	0,22
prostate-outcome	21	2	12600 (12600,0)	0,00%	0,32	0,25	0,21
alizadeh-lymph	96	9	4026 (4026,0)	5,09%	0,55	0,29	0,27
MLL-completo	72	3	12582 (12582,0)	0,00%	0,45	0,34	0,31
lungcancer-ontario	39	2	2880 (2880,0)	5,96%	0,46	0,37	0,33
DLBCLOutcome	58	2	7129 (7129,0)	0,00%	0,46	0,38	0,34
pomeroy-cns-dataset-C	60	2	7129 (7129,0)	0,00%	0,46	0,38	0,34
breastCancer-completo	97	2	24481 (24481,0)	0,00%	0,45	0,38	0,34
lung-harvard	203	5	12600 (12600,0)	0,00%	0,56	0,39	0,37
ma-2003-breast	61	3	1946 (1941,5)	0,30%	0,54	0,40	0,36
aml-all-completo	72	2	7129 (7129,0)	0,00%	0,48	0,40	0,36
DLBCLTumor	77	2	7129 (7129,0)	0,00%	0,49	0,41	0,37
leung-2002-gastric	126	3	6688 (6688,0)	5,87%	0,55	0,42	0,39
lung-Michigan	96	2	7129 (7129,0)	0,00%	0,51	0,44	0,39
prostate-tumorVSNormal	136	2	12600 (12600,0)	0,00%	0,52	0,45	0,40
lungCancer-completo	181	2	12533 (12533,0)	0,00%	0,55	0,48	0,44
ovarian-61902	253	2	15154 (15154,0)	0,00%	0,57	0,50	0,46
DLBCL-NIH-completo	240	2	7399 (7399,0)	10,30%	0,62	0,54	0,49
BCR-ABL-completo	327	2	12558 (12558,0)	0,00%	0,61	0,54	0,50
E2A-PBX1-completo	327	2	12558 (12558,0)	0,00%	0,61	0,54	0,50
Hyperdip50-completo	327	2	12558 (12558,0)	0,00%	0,61	0,54	0,50
T-ALL-completo	327	2	12558 (12558,0)	0,00%	0,61	0,54	0,50

Experimentos Realizados

As bases utilizadas nestes experimentos preliminares encontram-se descritas na Seção A.1. Na Seção B.1, são discutidas algumas métricas de densidade propostas nesta pesquisa e utilizadas na execução do primeiro experimento. Nas Seções B.2 e B.3 são discutidos os resultados obtidos nos experimentos preliminares 1 e 2, respectivamente.

A fim de analisar se os resultados são significativamente diferentes, aplicou-se o teste de Friedman (Friedman 1940), considerando um nível de significância de 5%, assim como explicado na Seção 4.2.

B.1 Métricas de densidade

É sabido a partir da teoria do aprendizado computacional que, dado um espaço de hipóteses (neste caso, definido pela *Random Forest*), é possível determinar a complexidade do conjunto de treinamento (tamanho) para um classificador convergir (com maior probabilidade) para uma hipótese bem sucedida (Mitchell 1997, Chap. 7). Isto requer conhecimento do tamanho do espaço de hipóteses (ou seja, sua cardinalidade) ou sua capacidade fornecida pela dimensão VC (Vapnik, Levin & Cun 1994). Na prática, encontrar o tamanho do

espaço de hipóteses ou a capacidade é difícil e apenas recentemente uma abordagem definiu a dimensão VC para árvores de decisão binárias, pelo menos parcialmente, uma vez que esta abordagem foi definida em termos de subárvores direta e esquerda (Aslan, Yildiz & Alpaydin 2009), enquanto que o padrão ouro deveria ser definido em termos do espaço de exemplos.

Por outro lado, métricas de bases de dados (espaço de exemplos) são muito menos discutidas na literatura. Nosso interesse é, uma vez que o espaço de hipóteses é fixado (mas seu tamanho ou sua dimensão VC são ambos desconhecidos ou infinitos), quais conjuntos de treinamento *parecem* ter um volume suficiente para que o aprendizado possa ser bem sucedido. Em um trabalho relacionado, algumas métricas de balanceamento de classe foram propostas (Perez & Baranauskas 2011). Uma vez que neste estudo foram usadas bases de dados com diferentes números de classes, exemplos e atributos, elas não podem ser agrupadas em algum sentido intuitivo usando essas três dimensões. Com base nisso, são propostas três diferentes métricas, mostradas em (B.1), (B.2) e (B.3), onde cada base de dados tem c classes, a atributos e n exemplos.

Estas métricas foram projetadas usando as seguintes ideias. Para um objeto físico, a densidade D é sua massa dividida pelo seu volume. Para uma base de dados, foi considerada sua massa como o número de exemplos; seu volume foi dado por seus atributos. Aqui considera-se o conceito de volume de um objeto (base de dados) como sua capacidade, ou seja, a quantidade de fluido (atributos) que o objeto pode conter, em vez de a quantidade de espaço que o objeto desloca. Sob estas considerações, tem-se $D \triangleq \frac{n}{a}$. Uma vez que, em geral, estes números variam consideravelmente, um melhor modo de olhar para eles foi usando ambos números na escala logarítmica natural, $D \triangleq \frac{\ln n}{\ln a}$ o qual nos leva a (B.1). Na próxima métrica, foi considerado que o número de exemplos (massa) torna-se rarefeito pelo número de classes, resultando em (B.2) e a última métrica engloba bases de dados vazias (sem exemplos) e bases sem o rótulo da classe (aprendizado não supervisionado).

$$D_1 \triangleq \log_a n \quad (\text{B.1})$$

$$D_2 \triangleq \log_a \frac{n}{c} \quad (\text{B.2})$$

$$D_3 \triangleq \log_a \frac{n+1}{c+1} \quad (\text{B.3})$$

Considerando a hipótese comum em aprendizado de máquina que $c \leq n$, é óbvio que, para cada métrica D_i , $D_i \geq 0$, $i = 1, 2, 3$. Considera-se que se $D_i < 1$, a densidade é baixa e *talvez* o aprendizado a partir dessa base pode ser difícil, sob o ponto de vista computacional. Caso contrário, $D_i \geq 1$, a densidade é alta e o aprendizado *pode ser* mais fácil.

B.2 Experimento 1

Neste experimento foram utilizadas duas medidas para analisar os resultados: a área média ponderada sob a curva ROC (AUC) e a porcentagem de atributos usados em cada *Random Forest*. Para avaliar o desempenho deste experimento, usaram-se 10 repetições de validação cruzada com 10-folds. A média de todas as repetições para uma dada floresta em uma determinada base foi tomada como o valor do desempenho (AUC e porcentagem) para o par.

Os valores de AUC obtidos para cada base (29 bases, descritas no Apêndice A.1) e cada número de árvores usadas na *Random Forest* são mostrados na Tabela B.1. Também são mostrados os valores da média e da mediana assim como o *rank* médio obtido no teste de Friedman. Média, mediana e o *rank* médio são apresentados para os seguintes grupos: todas as bases; somente as 8 bases com baixa densidade; e somente as 21 bases com alta densidade.

Como pode ser observado, em todos os grupos (todos/8 baixa-densidade/ 21 alta-densidade) a floresta com 4096 árvores tem o menor (melhor) *rank* de todos. Além disso, no grupo das 21 alta-densidade, pode-se observar que as florestas com 2048 e 4096 árvores apresentam o mesmo *rank*. Analisando o grupo usando todas as bases e as 8 com baixa densidade, pode-se notar que a floresta com 512 árvores tem um *rank* melhor do que a floresta com 1024 árvores, contrariando o esperado. Outro resultado interessante é que os valores de mé-

dia e mediana das bases com alta densidade para cada uma das três primeiras iterações, $L = 2, 4, 8$, são maiores do que os valores das bases com baixa densidade; o oposto é verdadeiro para $L = 16, \dots, 4096$. Isto pode sugerir que bases com baixa densidade, de fato, exigem um poder de expressão maior (florestas maiores) do que bases com alta densidade. Esta potência expressiva, naturalmente, pode ser expressa como o tamanho do espaço da Random Forest (hipótese) ou sua dimensão VC.

A fim de obter um melhor entendimento, os valores AUC também são apresentados nas Figuras B.1, B.2 e B.3 considerando todas as bases, somente as 8 com baixa densidade e somente as 21 bases com alta densidade, respectivamente. Como pode ser visto, nas Figuras B.1 e B.2, ambas média e mediana aumentam conforme o número de árvores aumenta, mas além de 64 árvores estas figuras não apresentam grandes mudanças. Na Figura B.3, a média e a mediana não apresentam maiores mudanças a partir de 32 e 16 árvores, respectivamente.

Com estes resultados pode-se observar um comportamento assintótico, onde um aumento nos valores do AUC são difíceis de se obter, mesmo dobrando o número de árvores dentro da floresta. Um modo de compreender este comportamento assintótico é computando a diferença entre o AUC de uma iteração com a próxima (por exemplo, entre 2 e 4, 4 e 8, etc.). Estes resultados são apresentados nas Figuras B.4, B.5 e B.6 para todas as bases, 8 baixa densidade e 21 alta densidade, respectivamente. Para esta análise, nós excluimos as diferenças dos AUC das bases que atingiram o valor de AUC igual a 99.99% antes de 4096 árvores (valores em negrito na Tabela B.1). Analisando esta tabela, podemos notar que usando todas as bases e as 8 com baixa densidade, as diferenças do AUC (média e mediana) entre 32 e 64 árvores na floresta estão abaixo de 1%. Considerando as 21 bases com alta densidade, estas diferenças estão abaixo de 1% entre 16 e 32 árvores na floresta e abaixo de 0.3% entre 32 e 64 árvores.

Analisando a Figura B.4 foi feito um ajuste dos valores de média e mediana por meio dos mínimos quadrados para curva $g = aL^b$, onde g representa a porcentagem da diferença do AUC (ganho) e L é o número de árvores dentro da floresta. Obteve-se, usando todas as bases e o valor da mediana da diferença do AUC $a = 6.42$ e $b = -0.83$ com coeficiente de correlação $R^2 = 0.99$ e usando a média da diferença do AUC $a = 6.06$ e $b = -0.65$ com coeficiente de correlação $R^2 = 0.98$. Para fins práticos, é possível aproximar para $g \simeq \frac{7}{L}\%$ com coeficiente de

Tabela B.1: Valores AUC, média, mediana e rank médio obtidos nos experimentos. Dados em negrito representam valores excluídos da análise da diferença do AUC.

Bases	Número de árvores											
	2	4	8	16	32	64	128	256	512	1024	2048	4096
GCM	0.72	0.77	0.83	0.87	0.89	0.91	0.91	0.92	0.92	0.92	0.93	0.93
Lymphoma	0.85	0.92	0.96	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99
CNS	0.50	0.52	0.56	0.58	0.59	0.59	0.59	0.58	0.60	0.60	0.60	0.60
Leukemia	0.76	0.85	0.93	0.97	0.98	0.98	0.99	0.99	0.99	0.99	0.99	1.00
Leukemia nom.	0.72	0.81	0.91	0.96	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Ovarian 61902	0.90	0.96	0.98	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
Lung Cancer	0.58	0.64	0.66	0.65	0.65	0.66	0.66	0.68	0.69	0.68	0.68	0.69
C. Arrhythmia	0.71	0.77	0.82	0.85	0.87	0.88	0.89	0.89	0.89	0.89	0.89	0.89
<hr/>												
Dermatology	0.97	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
HD Switz.	0.55	0.55	0.58	0.58	0.60	0.61	0.60	0.60	0.60	0.61	0.61	0.61
Lymphography	0.82	0.87	0.90	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
Hepatitis	0.76	0.80	0.83	0.84	0.85	0.85	0.85	0.85	0.86	0.85	0.86	0.86
HD Hungarian	0.80	0.84	0.86	0.87	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88
HD Cleveland	0.80	0.84	0.87	0.88	0.89	0.89	0.90	0.89	0.89	0.89	0.90	0.90
P. Patient	0.45	0.45	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
WDBC	0.96	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Splice Junction	0.87	0.93	0.97	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00
Heart Statlog	0.80	0.84	0.87	0.89	0.89	0.89	0.90	0.90	0.90	0.90	0.90	0.90
Allhyper	0.89	0.95	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Allhypo	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Sick	0.92	0.97	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Breast Cancer	0.60	0.63	0.64	0.65	0.65	0.66	0.66	0.67	0.66	0.66	0.66	0.66
Hypothyroid	0.95	0.97	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
ANN Thyroid	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
WBC	0.97	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
C. Method	0.62	0.64	0.66	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.68	0.68
Pima Diabetes	0.72	0.76	0.79	0.81	0.81	0.82	0.82	0.82	0.82	0.82	0.83	0.83
Liver Disorders	0.66	0.70	0.72	0.74	0.75	0.76	0.76	0.77	0.77	0.77	0.77	0.77
H. Survival	0.58	0.60	0.61	0.62	0.63	0.63	0.64	0.64	0.64	0.64	0.64	0.64
<hr/>												
Todas												
Média	0.77	0.81	0.84	0.85	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.87
Mediana	0.80	0.84	0.87	0.89	0.89	0.91	0.91	0.92	0.92	0.92	0.93	0.93
Rank médio	11.83	10.55	8.79	8.05	6.88	5.81	5.12	4.62	4.31	4.39	3.91	3.72
<hr/>												
8 baixa-densidade												
Média	0.72	0.78	0.83	0.85	0.87	0.88	0.88	0.88	0.88	0.88	0.89	0.89
Mediana	0.72	0.79	0.87	0.91	0.93	0.94	0.95	0.96	0.96	0.96	0.96	0.96
Rank médio	12.00	11.00	9.62	8.81	7.94	6.25	4.81	4.44	3.37	3.69	3.37	2.69
<hr/>												
21 alta-densidade												
Média	0.79	0.82	0.84	0.85	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
Mediana	0.80	0.84	0.87	0.89	0.89	0.89	0.90	0.90	0.90	0.90	0.90	0.90
Rank médio	11.76	10.38	8.47	7.76	6.47	5.64	5.24	4.69	4.66	4.66	4.12	4.12

correlação $R^2 = 0.99$, o qual indica que este é um bom ajuste também. Por exemplo, usando $L = 8$ árvores com AUC igual a 0.90, é possível estimar o AUC para 16 árvores (dobrando L), assim, $g \simeq \frac{7}{8}\%$ e o valor do AUC esperado para 16 árvores é $0.90 \times (1 + \frac{7}{8}) \simeq 0.91$. Certamente, esta fórmula pode ser usada com qualquer número positivo de árvores, por exemplo, considerando uma floresta com 100 árvores, o ganho esperado no AUC para uma floresta com 200 árvores é 0.07%.

Na Tabela B.2 são representados os resultados do teste *post-hoc* após o teste de Friedman e a rejeição da hipótese nula. Nela são mostrados os resultados usando todas as bases, as 8 com baixa densidade e as 21 com alta densidade. Nesta tabela Δ ($\blacktriangle$) indica que a *Random Forest* na linha especificada é melhor (significativamente) do que a *Random Forest* na coluna especificada; ∇ ($\blacktriangledown$) a *Random Forest* na coluna especificada é pior (significativamente) do que a *Random Forest* na coluna especificada; o indica nenhuma diferença. O triângulo inferior desta tabela não é mostrado pois ele apresenta resultados opostos ao triângulo superior por simetria.

Algumas observações importantes podem ser feitas a partir da Tabela B.2. Primeiro, pode-se observar que não há diferença significativa entre um dado número de árvores (2^j) e seu dobro (2^{j+1}), em todos os casos. Quando há uma diferença significativa, isto somente aparece quando é comparado o número de árvores (2^j) com pelo menos quatro vezes este número (2^{j+2}). Segunda observação importante é que a partir de $64 = 2^6$ uma diferença significativa foi encontrada apenas em $4096 = 2^{12}$, somente quando a *Random Forest* cresceu sessenta e quatro vezes. Terceiro ponto, a partir de $128 = 2^7$ árvores, não há mais diferença significativa entre as florestas até 4096 árvores.

Visando analisar a porcentagem de atributos usados, os boxplots deste experimento são mostrados nas Figuras B.7, B.8 e B.9 para todas as bases, as 8 bases com baixa densidade e as 21 bases com alta densidade, respectivamente. Considerando a Figura B.7, os valores de média e mediana da floresta usando 128 árvores corresponde a 80.91% e 99.64% dos atributos, respectivamente. Quando analisam-se as 8 bases com baixa densidade na Figura B.8, é possível notar que mesmo com 4096 árvores na floresta, nem todos os atributos foram usados. Porém, como pode ser visto, esta curva tem um formato diferente (sigmoidal) das demais curvas nas Figuras B.7 e B.9 (exponencial). Além disso, a curva sigmoidal parece crescer até

Tabela B.2: Resultados do teste de Friedman para os valores AUC usando todas bases/8 baixa densidade/21 alta densidade

[illegible]

seu máximo em 100%.

Figura B.1: Valor AUC em todas as bases de dados (29 bases)

Figura B.2: Valor AUC nas 8 bases com baixa densidade

Figura B.3: Valor AUC nas 21 bases com alta densidade

Figura B.4: Diferenças do valor AUC em todas as bases

Figura B.5: Diferenças do valor AUC nas 8 bases com baixa densidade

Figura B.6: Diferenças do valor AUC nas 21 bases com alta densidade

Figura B.7: Porcentagem de atributos usados em todas as bases

Figura B.8: Porcentagem de atributos usados nas 8 bases com baixa densidade

Figura B.9: *Porcentagem de atributos usados nas 21 bases com alta densidade*

Conclusão

Este primeiro experimento preliminar analisou o número de árvores para serem geradas por uma *Random Forest*. Os resultados obtidos neste experimento mostram que, as vezes, um número maior de árvores em uma floresta somente aumenta o custo computacional e não apresenta ganho de desempenho significativo. Eles também indicam que a média e mediana do AUC tende a convergir assintoticamente. Outra observação é que não há diferença significativa entre o uso de um número de árvores dentro de uma *Random Forest* e o seu dobro. A análise de 29 bases de dados mostra que a partir de 128 árvores não há mais diferença significativa entre as florestas usando 256, 512, 1024, 2048 e 4096 árvores. Os valores da média e mediana do AUC não apresentam maiores mudanças a partir de 64 árvores. Portanto, é possível sugerir, baseado neste experimento, um intervalo entre 64 e 128 árvores na floresta. Utilizando esses números de árvores é possível obter um bom balanço entre AUC, tempo de processamento e uso de memória. Também foi descoberta uma relação experimental (inversamente proporcional) para o ganho do AUC quando dobra-se o número de árvores em qualquer floresta. Analisando a porcentagem de atributos usados, pode-se notar que a mediana alcança o conjunto inteiro de atributos com 128 árvores na floresta. Assim, assintoticamente a tendência indica que a *Random Forest* vai usar todos os atributos e isso não é interessante em alguns casos, por exemplo em bases com muitos atributos (ou seja, bases de expressão gênica), uma vez que nem todos são importantes para o aprendizado do conceito (Netto, Nozawa, Mitrowsky, Macedo & Baranauskas 2010; Gamberger, Lavrac, Zelezny & Tolar 2004).

B.3 Experimento 2

Após a realização do Experimento 1 e a descoberta de que um intervalo entre 64 e 128 árvores em uma floresta é o mais indicado para uma estimativa de precisão, tentou-se gerar florestas contendo 128 árvores para verificar a frequência que os atributos aparecem nas raízes das árvores. Porém, os experimentos utilizando essa quantidade de árvores não apresentaram estabilidade no subconjunto dos atributos mais utilizados. Também foram testadas florestas utilizando $\frac{a}{2}$ árvores, novamente sem estabilidade, onde a é o número de atributos de uma base. Finalmente, florestas contendo a e $2a$ árvores apresentaram resultados estáveis. Com isso é possível observar que uma boa precisão pode ser alcançada rapidamente com 64–128 árvores; este ponto de vista vê a *Random Forest* como uma caixa preta. Porém, analisando fatores específicos dentro de uma *Random Forest*, ou seja, analisando a *Random Forest* como uma caixa branca, mais árvores podem ser necessárias para encontrar outras métricas que não desempenho.

A seguir, os resultados de cada etapa deste experimento serão apresentados. Todas as etapas deste experimento utilizam a posição do atributo (ou seja, o índice do atributo na base de acordo com o Weka (Hall, Frank, Holmes, Pfahringer, Reutemann & Witten 2009), o qual começa no zero) como seu ID. Para avaliar o desempenho, foi realizada validação cruzada com 10-folds. Foram utilizadas 14 bases das 29 descritas no Apêndice A.1, pois foram analisadas somente as bases que apresentavam número de atributos maior do que 25, a fim de melhor analisar a estabilidade dos atributos mais usados, uma vez que existiam bases com 3 atributos e assim, seriam geradas florestas muito pequenas (com no máximo 6 árvores).

Etapa 1. Nesta etapa do experimento, buscou-se pela frequência média dos atributos no nível da raiz, por exemplo, se os atributos aparecem uniformemente ou se existe um subconjunto deles que é mais frequentemente usado.

Duas medidas foram usadas na realização desta etapa: o número de vezes que um atributo estava entre os m atributos selecionados aleatoriamente pelas árvores no nível da raiz (*vezesSelecionado*) e o número de vezes que este atributo foi, de fato, escolhido para ser o nó raiz (*vezesRaiz*). Então, usou-se a razão entre eles ($Frequencia = vezesRaiz / vezesSelecionado$)

Figura B.10: Frequência dos 10 atributos mais usados em todas bases. O eixo x corresponde ao número do atributo e o eixo y corresponde à frequência. Embora todos os eixos y tenham ficado em um intervalo de 0 e 1, este intervalo varia em alguns gráficos para melhor visualização.

para analisar a frequência dos atributos. Após ordenar as frequências de todos os atributos de cada base, foi possível notar que elas apresentaram um comportamento exponencial. Na Figura B.10 somente as dez maiores frequências são mostradas. Há quatro linhas em cada gráfico representando a média e a mediana das frequências para florestas utilizando a e $2a$ árvores (ordenadas pelas frequências médias da floresta usando a árvores). A partir desta figura, é possível observar que em alguns casos há um único atributo que sobressai (por exemplo, nas bases Allhyper e Allhypo, ambas contendo poucos atributos) e em outros casos, há um subconjunto de atributos mais frequentemente usados (por exemplo, nas bases Lymphoma e Leukemia, ambas contendo um grande número de atributos).

Etapa 2. Após a realização da Etapa 1, verificou-se que haviam atributos que apresentavam frequências muito próximas. Por exemplo, suponha que existem três atributos na subconjunto de atributos mais usados: A, B e C. Assumindo que todos eles possuem a mesma frequência, porém o atributo A tem erro *out-of-bag* estimado igual a 0.90, B igual a 0.65 e C igual a 0.20. Neste caso, assume-se que o atributo C é o melhor do subconjunto, uma vez que seu desempenho é o melhor. A questão que surge é como estimar o erro *out-of-bag* para

Figura B.11: Frequência dos 10 atributos mais usados em todas bases usando o erro *out-of-bag* estimado.

um dado atributo. Sabe-se que o atributo do nível de raiz é o mais importante na árvore e portanto, pode-se assumir que ele determina o desempenho da árvore. Baseado nessa suposição, utilizou-se o erro *out-of-bag* da árvore quando o atributo α está no nível da raiz como uma estimativa do desempenho do atributo α .

Com estas modificações, foi realizada a segunda etapa do experimento, na qual as frequências foram alteradas para $Frequencia(1 - OOB)$, onde *OOB* corresponde ao erro *out-of-bag* médio de um atributo como explicado anteriormente. Os resultados deste experimento são mostrados na Figura B.11. Analisando os resultados, pode-se observar que em todas as bases a frequência tem um comportamento exponencial ou similar, mesmo nas bases que apresentaram um comportamento linear na primeira etapa. Deste modo, usando o erro *out-of-bag* estimado, há geralmente um único atributo que sobressai em cada base.

Etapa 3. Como mencionado anteriormente, *Random Forests* não super-ajustam, embora as árvores dentro delas crescem sem poda. Neste experimento, foi analisado o comportamento da pré-poda, uma vez que isto pode acelerar a indução da *Random Forest*. Para realizar a pré-poda das árvores da *Random Forest*, o parâmetro *minNum* foi usado. Tal parâmetro

Tabela B.3: Resultados do teste de Friedman para os valores AUC usando 128 árvores e considerando um nível de significância de 5%; rank médio para cada valor de *minNum* e a porcentagem do tempo médio de execução.

<i>minNum</i>	1	2	3	5	7	11	13	17	19	23
1	○	▽	▽	▽	△	△	▽	△	△	△
2		○	▽	▽	△	△	△	△	△	△
3			○	▽	△	△	△	△	△	△
5				○	△	△	△	△	△	△
7					○	△	▽	△	△	△
11						○	▽	△	△	△
13							○	△	△	△
17								○	△	△
19									○	△
23										○
Rank Médio	5.18	4.64	4.46	4.39	5.46	5.54	4.93	6.25	6.79	7.36
Tempo(%)	100.00	95.04	92.22	85.49	81.80	76.15	73.64	70.30	69.39	66.91

determina o número mínimo de exemplos em uma folha, onde o valor default na Weka (Hall, Frank, Holmes, Pfahringer, Reutemann & Witten 2009) é 1.0, o que gera árvores muito grandes. Baseado nisso, foram utilizados dez valores diferentes de *minNum*: 1, 2, 3, 5, 7, 11, 13, 17, 19 e 23. Como explicado na Seção B.2, para a estimação da precisão um intervalo de 64–128 árvores é suficiente. Desse modo, foram construídas florestas com 128 árvores. Para analisar os diversos valores de *minNum*, os valores AUC foram usados e o teste de Friedman (Friedman 1940) foi aplicado, considerando um nível de significância de 5%. Também foi observado o tempo médio de execução para induzir a floresta usando cada diferente valor de *minNum*. Esta medida foi utilizada com base no tempo médio de execução para induzir a floresta usando *minNum* = 1, ou seja, o tempo de execução para induzir esta floresta foi tomado como 100% e as porcentagens restantes foram calculadas baseadas nesta, visto que para valores maiores de *minNum* o tempo é menor, devido ao processo de pré-poda que interrompe o crescimento das árvores.

Na Tabela B.3 são apresentados os resultados do teste *post-hoc* após o teste de Friedman e a rejeição da hipótese nula, o *rank* médio e a porcentagem do tempo médio de execução de cada valor de *minNum*. Nesta tabela, a legenda é a mesma da Tabela B.2

Pode-se observar que o tempo de execução diminui conforme o valor de *minNum* aumenta, o que é esperado uma vez que valores maiores representam uma árvore menor e portanto, um tempo de execução mais curto. Embora não há diferenças significativas, é possível notar a partir da Tabela B.3 que *minNum* = 5 parece ser um valor interessante com o melhor *rank* médio. Usando este valor, os passos da segunda etapa foram repetidos e os

Figura B.12: Frequência dos 10 atributos mais usados em todas bases usando o erro out-of-bag estimado e $\minNum = 5$.

resultados são mostrados na Figura B.12. Como é possível notar, não houve diferenças significativas entre os comportamentos das frequências mostrados nas Figuras B.11 e B.12, mas este último é quase 15% mais rápido do que o primeiro. Contudo, houve diferenças em alguns subconjuntos dos dez atributos mais usados. Por exemplo, em quatro bases (Leukemia, Lymphoma, Ovarian e WDBC) os dez atributos mais usados foram os mesmos em ambas etapas, porém a sequência deles foi diferente; em outras quatro bases (Arrhythmia, CNS, Leukemia nom. e Lung Cancer) alguns atributos apareceram em ambas etapas (na mesma ordem e em ordem diferente) e houve alguns atributos diferentes entre elas. Por outro lado, em seis bases (Allhyper, Allhypo, Dermatology, Sick, Splice e Thyroid) as sequências dos dez atributos mais usados foram as mesmas em ambas etapas.

Conclusão

Este segundo experimento preliminar visou analisar a frequência que os atributos aparecem nas raízes das árvores de uma Random Forest. Foram construídas *Random Forests* usando a e $2a$ árvores, onde a é o número de atributos da base de dados. Analizando os resultados, pode ser observado que a *Random Forest* escolhe um subconjunto de atributos

mais utilizados ou um único atributo em cada base. Além disso, a frequência que os atributos aparecem na raiz tem um comportamento exponencial. Parece que quando usa-se a e $2a$ árvores, o subconjunto de atributos é estável. Pode-se também observar que nem sempre que um atributo é usado mais do que outro, seu desempenho é melhor. As vezes outro atributo apresenta um erro *out-of-bag* estimado menor e quando essa métrica é usada, este atributo fica a frente do primeiro. Usando o erro *out-of-bag* estimado como um complemento, pode-se notar que em todas as bases, um atributo sobressaiu. É interessante notar que em bases de dados de expressão gênica, encontrar um subconjunto ou um único melhor atributo pode facilitar a descoberta de conhecimento e melhorar o desempenho da classificação.